NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Enumeration of the Finite Projective Planes of Order Nine

Galina I. Kolesova

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

March, 1989

ISBN 0-315-51357-8

ABSTRACT

Enumeration of the Projective Planes of Order Nine

Galina I. Kolesova

Four non-isomorphic projective planes of order 9 are known. Using a computer search, we show that there are no more such planes. The solution to the problem of generating the projective planes, which are represented in terms of 91×91 incidence matrices, consists of four sequential steps: determination of all sets of 27 columns of the incidence matrices, extension of these columns into matrices of 40 columns, extension of these matrices into complete matrices, and identification of the planes obtained. The first step is reducible to the generation of 8×8 non-isomorphic latin squares. Their number is 283657, and this is the number of partial incidence matrices of 27 columns. Only 21 out of the 283657 latin squares extend to 40 columns. These give rise to 326 partial 40-column incidence matrices and 325 out of the above 326 extend to a complete plane. These planes are shown to correspond to four known isomorphism classes: the Desarguesian plane and the Hughes plane which are self–dual, and the left near-field and its dual, the right near-field, planes.

Acknowledgements

This thesis was completed with the constant help of my supervisor, Prof. Clement W. H. Lam. It is my pleasant duty to express here my most sincere gratitude to him — for his unfailing interest, his sound advice and his penetrating criticism, which ensured the joy and satisfaction I have drawn from this work.

I am also sincerely grateful to Prof. John McKay for his interest in the work as well as for reading and correcting the thesis.

I am very much indebted to Larry Thiel, whose programs I successfully used.

My thanks go as well to Stanley Swiercz (Dziekuje bardzo!) for friendly help at all times.

I am also grateful to Dr. Eric Regener for the final reading of the thesis and many useful comments he made.

And last but not least, many thanks to my family, who supported me and were more than patient throughout my studies.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

FIGURES

TABLES

INTRODUCTION

A *finite projective plane* of order n ($n \geq 2$) is a collection of n^2+n+1 lines and n^2+n+1 points such that

1) every line contains $n+1$ points,

2) every point lies on $n+1$ lines,

3) any two distinct lines intersect at exactly one point, and

4) any two distinct points lie on exactly one line.

The set of points is denoted by $\{p_1, p_2, ..., p_m\}$, the set of lines, by $\{L_1, L_2, ..., L_m\}$, where $m = n^2+n+1$, and the plane itself, by π_n.

As an example of a projective plane, let us consider π_2, a plane of order 2. π_2 contains 7 lines $\{L_1,..., L_7\}$ and 7 points $\{p_1,..., p_7\}$, each line contains 3 points and each point is an intersection of 3 lines. One possibility is to choose

$L_1 = \{p_1, p_2, p_4\}$, $L_2 = \{p_2, p_3, p_5\}$, $L_3 = \{p_3, p_4, p_6\}$, $L_4 = \{p_4, p_5, p_7\}$,

$L_5 = \{p_5, p_6, p_1\}$, $L_6 = \{p_6, p_7, p_2\}$, $L_7 = \{p_7, p_1, p_3\}$.

This plane can also be represented as in Fig. 1.

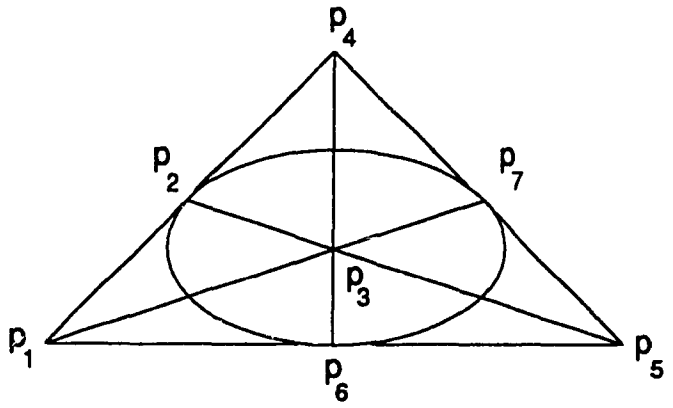

Fig. 1. Projective plane of order 2.

A plane is said to be *dual* to another if it can be obtained from the other by exchanging lines and points.

Two projective planes π, π' are *isomorphic* if one can be obtained from the other by a transformation α which preserves the incidence of the points and the concurrence of the lines, i.e., for every pair of points (p, q) lying on a line L, and for every pair of lines (L, M) intersecting at a point p, there is a transformation α such that $\alpha(p) = p'$, $\alpha(q) = q'$, $\alpha(L) = L'$, $\alpha(M) = M'$, and p', q' lie on L', and L' and M' are intersecting at p'. α is said to be an isomorphic transformation or, for short, an isomorphism of π. For finite projective planes, the only isomorphic transformations are point and line relabellings. An isomorphism α of a plane into itself is called a *collineation* or *automorphism* .

In the following we review the main results concerning the projective planes and, in particular, projective planes of order 9. We will not give any definitions in this part since the purpose of this review is to show how our results correlate with the known ones.

Projective planes do not exist for all orders. The Bruck–Ryser Theorem ([6], p. 394) states that if $n \equiv 1, 2 \pmod 4$, there cannot be a plane of order n unless n can be expressed as a sum of two integral squares, $n = a^2 + b^2$. A consequence of this theorem is that no projective plane of order 6 exists. There is exactly one plane of each order from 2 to 8 excluding 6. These are all Desarguesian planes, a property of which is that they can be embedded in a projective 3–space ([6], p. 394).

For $n = 9$, it is known that projective planes do exist, and four such planes were found at the beginning of this century. One of these planes is Desarguesian. Two others, the left near-field and its dual, the right near-field plane, are coordinatized by the Veblen-Wedderburn system (they are obtained using a special coordinatization, based on the lines of a plane, and defining algebraic operations on the system of coordinates. The algebraic

system introduced is a ternary ring and the Veblen-Wedderburn system is a special case.). The Hughes plane, found by Veblen and Wedderburn in 1907, is called so because it was later (1957) generalized by Hughes ([8]), who has shown that this plane is non-Desarguesian and cannot be obtained by coordinatization using a Veblen-Wedderburn system. A description of these planes can be found in books on projective planes, for example, Hall [6], Hughes [8], or Stevenson [16].

There have been many attempts to determine whether the above list of planes for order 9 is complete. Most searches were for planes with some special properties, such as having an elementary abelian addition in an appropriate ternary ring, which is a property shared by the four known planes. Hall *et al.* [7], Killgrove [9], Killgrove and Parker [10], and Parker *et al.* [15] show that there are no more planes of order 9 with this property. Another approach calls for determination of the properties which the planes, if they exist, must have. Methods of coding theory have been applied to find corresponding properties of the codes for any plane of order 9 (see Hall [5]). But no answer on whether the list of the four planes of order 9 is complete was provided.

The present work aims at the generation of all the projective planes of order 9 to within isomorphism, i. e. a direct constructive approach necessarely using computers. The problem was thus stated by Lam after several years of searching for projective planes of order 10 (see Lam *et al.* [14]). Our work became possible due to the above (see Lam *et al.* [13], [17]), which led to a series of research with regard both to the theoretical framework and computer implementation.

The planes resulting from the generation are exactly the four planes which are already known.

We describe here the algorithm used and the results obtained. Chapter 1 contains the mathematical preliminaries concerning projective planes, and, in particular, projective

planes of order 9. Chapter 2 of the thesis concerns the algorithm used. The planes in the work are represented using incidence matrices of order 91×91, 27 columns of which are mapped onto a latin square of order 8. Chapter 3 contains a description and results of the first part of the algorithm – the generation of the 8×8 latin squares. Chapter 4 contains the results of the extension of 27 columns of an incidence matrix into complete planes. This part is divided into two: first, the 27 columns are extended into 40 columns of an incidence matrix and then into complete incidence matrices. The resulting matrices, or the planes they represent are not all non–isomorphic. The last part of the algorithm concerns the identification of the planes: all the planes are transformed into a canonical form which allows us to determine the classes of non–isomorphic planes. Chapter 5 contains the results of the identification. And, finally, the Conclusion presents an analysis of the results obtained. In addition, computer listings of some results are included in Appendices.

CHAPTER 1. Planes. Definition and Representation.

In this chapter we describe the representation of a projective plane by an incidence matrix, give all necessary definitions for projective planes and describe one-to-one mapping of 27 columns of an incidence matrix onto an 8×8 latin square.

1.1. Representation in Terms of Incidence Matrices.

We represent a projective plane π_n of order n by an square *incidence matrix* A of size (n^2+n+1). If point p_j lies on line L_i, then $a_{ij} = 1$ and $a_{ij} = 0$ otherwise, thus the columns of the matrix represent the points and the rows represent the lines of the plane.

For example (see Fig. 1), an incidence matrix of the plane of order 2 is as shown in Fig. 2.

1	1	0	1	0	0	0
0	1	1	0	1	0	0
0	0	1	1	0	1	0
0	0	0	1	1	0	1
1	0	0	0	1	1	0
0	1	0	0	0	1	1
1	0	1	0	0	0	1

Fig. 2. Incidence matrix of the plane of order 2.

In terms of incidence matrices, a {0,1} square matrix A of size n^2+n+1 represents a projective plane of order n if

1. each row of A contains exactly $(n + 1)$ ones,

2. each column of A contains exactly $(n + 1)$ ones,

3. the inner product of any two distinct rows of A is equal to 1, and

4. the inner product of any two distinct columns of A is equal to 1.

Projective planes are indistinguishable under point and line relabellings. Since such relabellings correspond to column and row permutations in the incidence matrix, the

definition of the isomorphism of planes in terms of incidence matrices is equivalent to the following statement:

Two projective planes of order n are *isomorphic* if and only if their incidence matrices can be obtained one from the other by independent row and column permutations. An automorphism, or *collineation,* of a plane is a matrix transformation which consists of independent row and column permutations preserving the incidence matrix. The set of all automorphisms is a group, called the automorphism group.

1.2. Standard Form of 27 Columns.

Let A be a 91×91 incidence matrix of a projective plane of order 9. We will show that, by applying the row and column permutations (the isomorphism transformations for the planes), the incidence matrices can be transformed into a special form, called a *standard form*; the first 27 columns of it are shown in Fig. 3. The first three columns of the standard form represent non-collinear points (let them call a *triangle*). Note that there are

$$\frac{91\times90\times(89-8)}{3!} = \frac{91\times90\times81}{3!}$$

choices for the triangles (8 out of 89 is the number of points collinear to the first two chosen points; they lie in columns 12 : 19).

We do not give a formal description of the standard form assuming that the picture given on Fig. 3 is clear and self–explaned.

Below we will show that as long as an triangle is chosen, the standard form of 27 columns of the incidence matrix is defined uniquely within column and row permutations.

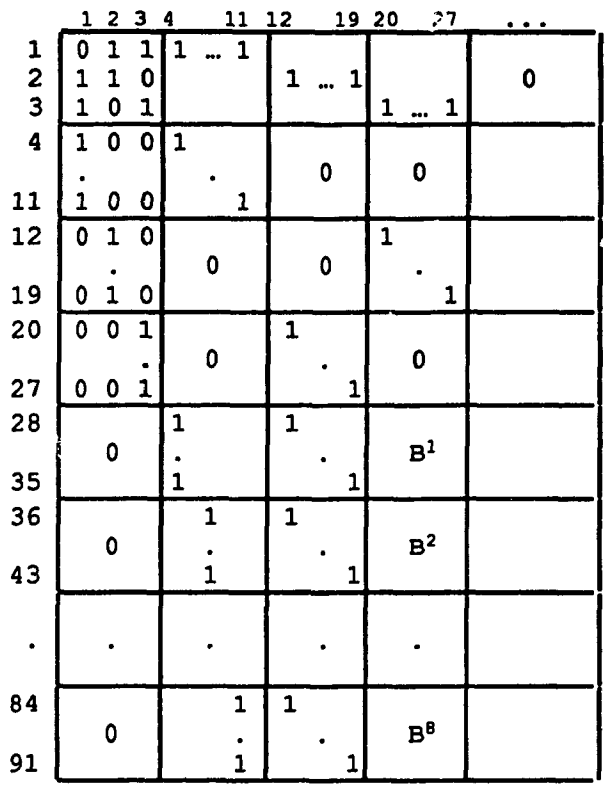

Fig. 3. The standard form of 27 columns.

Let us start with the notation:

$A = \{a_{ij}\}$, where $i, j = 1, ..., 91$, is an incidence matrix.

A^i stands for column i of A, and A_i for row i of A.

$A_{i:j}$ stands for the *row block*, containing rows i through j inclusive.

$A^{i:j}$ stands for the *column block*, containing columns i through j inclusive.

$A^{i:j}_{k:s}$ stands for the $u \times v$ block, $\{a_{rt}\}_{k \le t \le s}^{i \le r \le j}$, where $u = i - j + 1$, $v = s - k + 1$.

The column blocks $A^{4:11}$, $A^{12:19}$, and $A^{20:27}$ have special names:

$A^{4:11}$ - *row index block*,

$A^{12:19}$ - *relabelling block*, and

$A^{20:27}$ - *column index block*.

$A_k \supset A^i \cap A^j$ means that the intersection of vectors A^i and A^j has 1 in the k-th position, i. e., $a_{ki} = a_{kj}$.

$B^k = A^{20:27}_{20+8k:27+8k} = \{b^k_{ij}\}_{j \le 8}^{i \le 8}$; B^k is said to be block k of the column block $A^{20:27}$.

The structure of the standard form implies the following three properties of the entries of B^k:

1. Every row A_{28} through A_{91} can intersect row A_3 only in $A^{20:27}$; therefore, every B^k must have exactly one 1 in every row, or every pair (k, i) uniquely defines j such that $b^k_{ij} = 1$, where $i, j, k = 1,..., 8$.

2. Every column A^{19+k} ($k = 1,..., 8$) can intersect column A^{3+k} only in row block $A_{20+8k:27+8k}$; therefore, every B^k must have exactly one 1 in every column, or every pair (k, j) uniquely defines such i that $b^k_{ij} = 1$, where $i, j, k = 1,..., 8$.

3. If $b^k_{ij} = 1$, then column A^{19+j} intersects A^{11+i}; therefore, there are no more $k' \neq k$ such that $b^{k'}_{ij} = 1$, or every pair (i, j) uniquely defines k such that $b^k_{ij} = 1$.

From the above we deduce that the positions of ones in blocks $B^1,..., B^8$ define triplets (i, j, k) such that

B1. each pair (i, j), (i, k), and (j, k) takes all 64 possible values (1,1),...,(8,8), and

B2. every pair of values uniquely determines the third of the triplet.

Proposition 1.2. A triplet (i, j, k) with the above properties define a latin square of order 8.

To prove this (which, in fact, is known as one of possible definitions of a latin square) let us first give the definition of a latin square.

A *latin square* of order n is a $n \times n$ matrix satisfying the following properties:

L1. all the entries are integers between 1 and n,

L2. in a row, no entry is repeated,

L3. in a column, no entry is repeated.

Proof. L1 is obvious by the construction.

L2 is satisfied, otherwise, if there are two $j \neq j'$ such that $(i, j, k) = (i, j', k)$ a pair (i, k) would define two different j (this contradicts to the definition of the triplet). Similarly L3 can be proved.

1.3. Mapping of 27 Columns onto an 8×8 Latin Square.

Since each incidence matrix has more than one standard form, a natural question is how the isomorphic transformations, or equivalently, the row and column permutations of the incidence matrix affect the latin square which represents the columns 20 through 27 of the standard form.

Proposition 1.3. The choice of the first three columns uniquely defines a latin square within the following transformations: row and column permutations, relabelling of its entries, inverse permutation applied to all rows simultaneously, and matrix transposition.

Proof. We shall prove this in two steps:

I: we fix the choice and the order of the first three columns, and

II: we permute the first three columns given.

I. Let us fix not only the choice of the first three columns, but also their order. This implies the following:

a) $A_{1:3}^{1:3}$ is uniquely defined by forcing $A_{1:3}^{1:3}$ to the form shown in Fig. 3;

b) The column blocks $A^{4:11}$, $A^{12:19}$, $A^{20:27}$ are uniquely defined, since the rows A_1, A_2, A_3 each must have eight ones in $A^{4:11}$, $A^{12:19}$, and $A^{20:27}$ (see Fig. 3)

c) Similarly, the row blocks $A_{4:11}$, $A_{12:19}$, $A_{20:27}$ are uniquely defined, since the columns A^1, A^2, and A^3 must each have eight ones in $A_{4:11}$, $A_{12:19}$, and $A_{20:27}$ respectively (see Fig. 3),

d) Now the incidence matrix can be put into the standard form by the following transformations:

- permute the rows of $A_{4:11}$ to reduce $A_{4:11}^{4:11}$ to the identity matrix;
- permute the columns of $A^{4:11}$ to group the ones in row blocks $A_{28:34}$,..., $A_{84:91}$;
- permute the rows of every row block $A_{20:28}$ through $A_{84:91}$ to reduce the 8×8 blocks in $A^{12:19}$ to identity matrices;
- permute the columns of $A^{20:27}$ or the rows of $A_{12:19}$ to reduce $A_{12:19}^{20:27}$ to the identity matrix.

As we have shown, the choice and the order of the first three columns of any incidence matrix define the columns of blocks $A^{4:11}$, $A^{12:19}$, $A^{20:27}$. Therefore, permutations preserving the choice and the order of the first three columns are restricted by the independent column permutations inside the blocks $A^{4:11}$, $A^{12:19}$, and $A^{20:27}$. The row permutations are implied by the column permutations in order to preserve the standard form. Let A be an incidence matrix in the standard form. Then a list of all possible transformations preserving the standard form consists of the following three types of permutations:

T1. A column permutation applied within the column block $A^{20:27}$. This permutation implies the row permutation in the row block $A_{12:19}$ in order to restore the identity of $A_{12:19}^{20:27}$.

T2. A column permutation applied within the column block $A^{12:19}$. This permutation implies the same row permutation within all row blocks $A_{20:27}$, ..., $A_{84:91}$ in order to restore the identity form of the 8×8 blocks in $A^{12:19}$.

T3. A column permutation applied within the column block $A^{4:11}$. This permutation implies the row permutation in the row block $A_{4:11}$ in order to restore the identity of $A^{4:11}_{4:11}$.

II. Let us permute the first three columns. Any of these permutations can be expressed by one of the following two transformations:

T4. The interchange of A^2 and A^3, the second and the third columns. To restore the order of the block $A^{1:3}_{1:3}$, rows A_2 and A_3 must be also interchanged. This implies that the column blocks $A^{12:19}$ and $A^{20:27}$ as well as the row blocks $A_{12:19}$ and $A_{20:27}$ must also be interchanged. These transformations reduce the incidence matrix under consideration to the form in which the row index block $A^{4:11}$ is in a standard form as well as the row blocks $A_{4:11}$, $A_{12:19}$, $A_{20:27}$. The relabelling block $A^{12:19}$ and column index block $A^{20:27}$ (starting from row 28) are interchanged as is shown in Fig. 4.

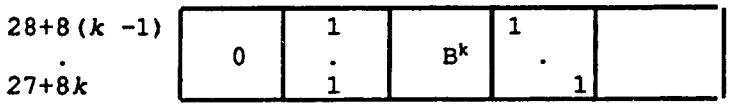

Fig. 4. Row block k. Effect of interchanging A^2 and A^3

To reduce the incidence matrix obtained to the standard form, it is sufficient to permute the rows inside each row block independently as follows:

$(i_1,..., i_8) \longrightarrow (1,..., 8)$, where $b^k_{i_t s} = 1$, and $s, t = 1,..., 8$.

The operation of the above type which maps a permutation $(i_1,..., i_8)$ onto the identity (1,..., 8) is called an *inverse permutation*. In our case the inverse permutation must be applied to all row blocks independently in order to reduce the matrix to the standard form.

T5. The interchange of A^1 and A^3, the first and the third columns. To restore the order of $A_{1:3}^{1:3}$ rows A_1 and A_3 must be interchanged. This implies that the column blocks $A^{4:11}$ and $A^{12:19}$ as well as the row blocks $A_{4:11}$ and $A_{20:27}$ must also be interchanged. These transformations reduce the incidence matrix under consideration to the form in which the column index block $A^{20:27}$ is unchanged, as well as the row blocks $A_{4:11}$, $A_{12:19}$, $A_{20:27}$. The row index block $A^{4:11}$ and the relabelling block $A^{12:19}$ (starting from row 28) are interchanged as is shown in Fig. 5.

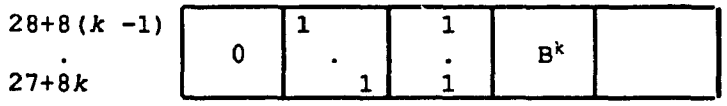

Fig. 5. Row block k . Effect of interchanging A^1 and A^3.

To reduce the incidence matrix obtained to the standard form, it is sufficient to permute the rows A_{28},..., A_{91} as follows: all the first rows of the row blocks $A_{28:35}$, ..., $A_{84:91}$ are combined into the first row block, the second rows of the row blocks into the second block, and so on.

All other permutations of the first three columns may be derived from T4 and T5.

As mentioned, the set of triplets (i, j, k), where $b_{ij}^k = 1$, defines an 8×8 latin square. Let us assume that the (k, j)-th entry of the latin square is i , i. e., the row index k of the latin square corresponds to the k-th row block $A_{28:34}$,...,$A_{84:91}$, the column index j corresponds to the j -th column of the column block $A^{20:27}$, and the i -th entry of the latin square corresponds to the i -th row index of an appropriate row block $A_{28:34}$,..., $A_{84:91}$.

Therefore, any standard form of the first 27 columns of a plane with a fixed set of the first three columns can be translated into a 8×8 latin square V to within the following transformations:

T1, a permutation of $A^{20:27}$ is equivalent to the column permutation of V;

T2, a permutation of $A^{12:19}$ is equivalent to the row permutation of V;

T3 , a permutation of $A^{4:11}$ is equivalent to the element relabelling of V;

T4, the interchange of A^2 and A^3 is equivalent to inverse permutation applied simultaneously to all rows of V;

T5, the interchange of A^1 and A^3 is equivalent to transposing V.

Conversely, by the construction, each latin square can be mapped into one 91×27 matrix. In Chapter 3, we describe Part I of our algorithm — the generation of all 8×8 latin squares isomorphic under the following matrix transformations:

ColP: Column permutation;

RowP: Row permutation;

EntR: Entry relabelling;

RowI: Inverse permutation applied to all the rows simultaneously; and

MatT: Matrix transposition.

CHAPTER 2. Description of the Algorithm.

The problem of generating the planes of order 9 is equivalent to generating the 91×91 incidence matrices to within independent row and column permutations. In this chapter we list the main steps of the algorithm, outline the description of the algorithm for each step, and list the computer programs written for the generation of the planes.

2.1. Structure of the Algorithm.

We reject a straightforward generation of the 91×91 incidence matrices because of the size of the problem. The problem has been divided into the following three parts:

Part I. Generation of 91×27 partial incidence matrices.

In any plane the rows and columns of the incidence matrix can be permuted in such a way that the first 19 columns are identical for all the planes. Columns 20 through 27 can then be mapped into an 8×8 latin square. This part thus consists in the generation of all non–isomorphic 8×8 latin squares.

Part II. Extension of 8×8 latin squares into complete planes.

This is accomplished in two steps.

IIa: the 91×27 partial incidence matrices are extended into 91×40 partial incidence matrices.

IIb: The 91×40 partial incidence matrices are extended into complete incidence matrices. This part uses an already existing program written by Larry H. Thiel.

Part III. Identification of the resulting planes.

The incidence matrices of the planes are reduced to canonical forms which we call certificates and which uniquely define the classes of isomorphic planes; these matrices are then compared element by element.

In the following we abbreviate "91×27 (91×40) partial incidence matrices" by "91×27 (91×40) incidence matrices" or "27 (40) columns".

2.2. Outline of the Algoritm.

To achieve the goals above, we use the well–known method of *backtracking*, or recursive depth–first search. We assume the reader is familiar with the main concepts, such as the *search tree* and the *levels* of the search.

Depending on the goal, we implement several modifications.The implementations of the method, or algorithms, are of two major types: algorithms for generating objects and algorithms for finding the canonical form of a given object. Thus, an algorithm of the first type, or GENERATION ALGORITHM, is used in Part I for generation of the 8×8 latin squares and in Part IIa for generation of the 91×40 incidence matrices. An algorithm of the second type, called a TRANSFORMATION ALGORITHM, is used in Part III to find the canonical form of a 91×91 incidence matrix, and in Part I to check whether an object, here a latin rectangle (the first k rows of a latin square), is in its canonical form. The generation algorithm is problem dependent, therefore two programs implementing the algorithm are written for the two above listed problems. The transformation algorithm maps all the operations defined for an object onto a symmetry group (this part of the algorithm is problem dependent), and determines the automorphism group of the operations of a considered object (which is done by interfacing with the problem independent part of the algorithm). The problem independent part of the algorithm is a package of runtime library routines. The package is written by Lam and Thiel [13, 17].

The two types of algorithms are used separately as well as together: the generation algorithm in Part IIa, the transformation algorithm in Part III, and both in Part I. In case of Part I the interaction of the algorithms occurs as follows: when the generation algorithm creates an object, the transformation algorithm is applied at several levels of the computational process to determine whether a partially computed object needs to be extended, i. e. whether it can give a rise to a new, non–isomorphic, complete object.

Let's start with the terminology used.

For the generation algorithm of Part I, *a search object* is an 8×8 latin square and for the generation algorithm of Part IIa, a 91×40 incidence matrix. An object is an ordered set of variables: the *rows* $(r_1,...,r_8)$ for the square and the *columns* $(c^{28},...,c^{40})$ for the matrix. On the i-th level of the search the $(i - 1)$ previous variables have been chosen, and the choice of the i-th variable is computed. The set D_i of all possible values of the i-th variable, or *candidates* for the i-th variable, depends (1) on the defining properties of an object, (2) on additional properties of an object, such as a canonical form in which the object is represented for the purpose of the algorithm, as well as (3) on the $(i - 1)$ previously chosen variables. For instance, (1) an defined property of an 8×8 latin square is that D_i $(i = 1,...,8)$ must be a subset of the permutations on $\{1,2,...,8\}$; (2) if we search for the latin squares in reduced form (where elements of the first row and the first column occur in natural order), then the candidates for row i must all begin with i, and (3) if seven rows of a latin square are all defined then no more than one element can be in D_8.

Thus, our backtracking algorithm in its pure form actually consists of the following: given D_1, for each i do the following: (B1) choose the next candidate for variable i from D_i; (B2) build D_{i+1}.

We shall describe an implementation of the method by giving a description of an object, the restrictions on the candidates for each variable, and the algorithm for maintaining the D_i's during the computational process.

Since the search trees in our problem are too big to permit the application of a pure backtracking algorithm, two methods are used to speed up the process: the first one is to minimize the size of the search tree by pruning the partial solutions which cannot give rise to new complete solutions; the second is to speed up the process of computing D_i. The former is used in Part I of the project, and the latter in Part IIa. Both methods are briefly outlined below.

Among all the solutions satisfying the problem, only non–isomorphic ones need to be chosen. When there are not too many solutions (as is the case with the 91×40 incidence matrices in Part IIa), the complete list of solutions can be created, and then a transformation algorithm can be applied to find the non–isomorphic solutions. But when the number of solutions is too big, as with the 8×8 latin squares, the transformation algorithm must be applied at some intermediate levels of the generation algorithm to *prune* the partial solutions which do not lead to new non–isomorphic complete objects.

From a user's point of view, the interaction with the package which implements the problem independent part of the transformation algorithm can be outlined as follows.

The set of all transformations applicable to a given object is mapped into the corresponding symmetry group: the set of independent row and column permutations of a $n \times m$ matrix is mapped onto the direct product of two symmetric groups $S_n \times S_m$. The specification of the group is passed to the package, and control is transferred to it by activating its main routine. The algorithm of the package uses a backtracking method in order to determine the automorphism group of the transformations applicable to a considered object (i. e. the set of the generators, the orbits, and the order of the group). For this purpose, when a new element of the group is chosen, the package asks the application program, the problem dependent part of the algorithm, what the operation corresponding to this element does to the object: it may transform the object into itself, or 'improve' it (minimize it in some sense), or make it 'worse'. Using this information, the

automorphism group of the object under the considered transformations is computed by the package. The transformation algorithm determines the orbits and the order of the automorphism group. The process of applying the transformation algorithm can be aborted in the middle if we are interested in knowing only whether a given object is canonical. The order of the automorphism group of the objects in question can be used in particular to determine the total number of objects (if the complete list of non-isomorphic classes is known) as well as for an internal consistency check of the computation. Lam and Thiel [13] propose an algorithm for this type of check which is used in Part I.

The second method used to speed up the process is a modification of the algorithm based on the use of a compatibility matrix. This modification has been developed by Thiel *et al* [17]. Its main idea is as follows.

The D_i's, $i = i_{first}, \ldots, i_{last}$; $N = i_{last} - i_{first} + 1$ ($i = 28,\ldots,40$ and $N = 13$ for Part IIa) are computed for all i so that (1) their elements do not conflict with the initial state of the object under consideration (the 27 columns for Part IIa), and (2) their elements satisfy to some additional properties based on the representation of an element required by the algorithm. Note that no assumptions are made about the candidates chosen for the previous variables (28, 29,..., i -1 for Part IIa).

We enumerate all the candidates from 1 to K, where K is the total number of all the elements of all the D_i's. Let x_i stand for the x-th candidate from the D_i-th set and y_j for the y-th candidate from the D_j-th set. Then the x_i and y_j are said to be *compabible* if the assignment $x_i \rightarrow i, y_j \rightarrow j$ is part of a possible object. The *compatibility matrix* C is defined as a $K \times K$ binary matrix where entry (x_i, x_j) shows the pairwise compatibility of candidates x_i and y_j.

During the computational process another matrix, *LIVE*, whose order is $N \times K$, is maintained as follows. The rows of *LIVE* have the same structure as the rows of the compatibility matrix, but its (i, y_j)-th entry ($i < j$) shows the compatibility of candidate y_j

with all chosen variables up to the i-th one. When the first i variables are chosen during the computational process, row i of *LIVE* may be computed (partially or completely) using $LIVE_{i-1}$ and C_{x_i}, where $LIVE_{i-1}$ is row $(i - 1)$ of *LIVE*, x_i is a candidate for variable i, and C_{x_i} is row x_i of matrix C. The variable $(i+1)$ can be chosen using only the row $LIVE_i$. The maintenance of the *LIVE* matrix can be implemented by fast bitwise computer operations, and therefore, the usage of this modification significantly speeds up the application of the backtracking method.

The use of the compatibility matrix has one more advantage: when a candidate for variable i is chosen and row i of *LIVE* is computed, one can look through its entries to find out whether there is at least one candidate for all the remaining variables. If not, the chosen candidate for variable i can be rejected, since it does not lead to a complete solution. This type of control, a *lookahead* method, can be applied partially, i. e., the computation of row i (and so the "lookahead") can be done in advance for a part of the row. Part IIa of the project implements the backtracking method with the above modification.

2.3. Computer Implementation of the Algorithm.

The algorithm is implemented as six programs written in PASCAL and run on VAX under VMS. These programs are:

1. "ROW3" creates the files (four), which are used as the input for "LSCONSIST". The files contain the lists of candidates for row 3 of the 8×8 latin squares which are compatible with all possible (four) sets of precomputed 2×8 latin rectangles.

2. "LSCONSIST" generates the 8×8 non-isomorphic latin squares. The input of the program is precomputed by "ROW3". The output is the list of the non–isomorphic latin squares stored in a compact form. A small part of the program with a different parameter is

used to decode the compact form of the result into a sequence of files with 1000 latin squares each in ASCII representation.

3. "EXPAND40" generates the 91×40 incidence matrices. The input is a file of latin squares in ASCII representation. The output is the list of latin squares that have extensions to 40–column partial incidence matrices and the 91×40 matrices themselves.

4. "MAKE_LARRY_INPUT" transforms a 91×40 matrix obtained by the third program, "EXTEND40", into the form required by the program "PLANE91" (written by Thiel), which computes all the complete incidence matrices.

5. "PL9CERT" transforms a complete incidence matrix obtained by "PLANE91" into canonical form.

6. "IDENT" compares the canonical form of an incidence matrix with those of the four known planes.

Besides these six programs which are the parts of the algorithm there is one more program, "FIND_CERT", to compute the canonical form of a given 8×8 latin square which has been written for checking purposes.

The following program and packages, developed by Lam and Thiel, are also used in the project:

1. "ISOMSHARE" is the implementation of the transformation algorithm.

2. "CONSIST" performs an internal consistency check.

3. "PLANE91" attempts to extend an initial incomplete incidence matrix in all possible ways to an incidence matrix of a given size (complete or incomplete). This program is used to extend a given a 91×40 matrix into complete incidence matrices.

Two of the above programs are time consuming: "LSCONSIST" and "EXPAND40". LSCONSIST takes 12 days on a VS3200 VAX system (and without consistency checking it takes 9 days); "EXPAND40" has run on a network of three VAXs

(two VS3200 and one VS2000 systems), concurrently. The total time elapsed is approximately 15 days.

The source programs together with the results are stored on a tape and can be obtained by request to the author.

CHAPTER 3. 8×8 Latin Squares. Definition and Generation.

The following three results on the enumeration of 8×8 latin squares are known (the necessary definitions will be given in the next section):

1. The number of the reduced latin squares was found by Wells [13]. It is equal 535,281,401,856. This result has been confirmed by Bammel and Rothstein [2] .

2. The number of isotopy classes was computed by Brown [3]. It is equal 1,676,257 . This result is widely known, but has not been confirmed. The latin squares, the representitives of the isotopy classes, have been classified by parity of the pairwise row permutations.

3. The number of main classes was computed by Arlazarov *et al* . [1]. It is equal 283,640. This result is also not confirmed. The latin squares, the representitives of the main classes, have been classified by the order of their automorphism group.

Since the latin squares of a main class are mapped onto a set of 27–column of an incidence matrix of a plane, we had to repeat similar computations to obtain a list of representatives of the main classes.

In this chapter we give the definition of latin squares following Dénes and Keedwell [4] and then de · ibe the algorithm for their generation.

3.1. Definitions and Notations.

We repeat the definition of a latin square and the set of operations under which we consider the latin squares to be isomorphic.

A *latin square* of order n is a $n \times n$ matrix satisfying the following properties:

1. all the entries are integers between 1 and n ,

2. in a row, no entry is repeated,

3. in a column, no entry is repeated.

The set of isomorphic operations on latin squares:

ColP: Column permutation;

RowP: Row permutation;

EntR: Entry relabelling;

RowI: Inverse permutation applied to all the rows simultaneously; and

MatT: Matrix transposition.

A partial latin square containing less than n rows is said to be a *latin rectangle* .

A *reduced* latin square is one in which elements 1, 2, ..., n of its first row and column occur in natural order.

Two latin squares are *isotopic* if one can be obtained from the other by permuting the rows, permuting the columns, and relabelling its elements, (**ColP, RowP, EntR** transformations). Isotopy is an equivalent relation partitioning the set of all latin squares of order n into disjoint classes, called *isotopy,* or *equivalence, classes* . The order of the group of isotopy operations is equal to $(n\,!)^3$, where the three $n!$ terms represent the number of row permutations, the number of column permutations, and the number of element relabellings, respectively. Note that **ColP, RowP, EntR** are also applicable to latin rectangles. The isotopy operations which transform a latin square into itself form its *automorphism* group.

Two latin squares are *conjugates* if one can be obtained from the other by a combination of the inverse operation applied simultaneously to all rows (or to all columns), and the matrix transpose operation (**RowI, MatT** transformations). If we consider a latin square as a triplet (i, j, k), where i is an entry, j is the column index, and k is the row index, the *conjugacy operations* (i. e. those which produce conjugate

squares) are the permutations of the coordinates (i, j, k). Therefore, the set of conjugacy operations is isomorphic to the symmetry group S_3. The number of distinct conjugates is a divisor of 6.

A *main class* comprises all elements of an isotopy class together with their conjugates. The order of the group of operations which fix a main class is equal to $3! \times (n!)^3$, the order of an isotopy class multiplied by the order of S_3.

3.2. Description of the Generation Algorithm.

We consider a latin square of order n as an ordered set $\{r_1,...,r_n\}$, where r_i represents row i of the latin square, and is, in its turn, a permutation of $\{1,...,n\}$. The main concept of the generation algorithm is that of a precedence order of the objects (the latin squares, in this case). To establish such an order among the latin squares which we generate, we start with establishing an order among the permutations. For this purpose, we need to introduce some definitions and facts from combinatorics.

Several representations of permutations are known. We use the following three:

1. A permutation $p(k) = i_k$ represented by its image $\{i_1 i_2...i_n\}$ is used to denote a row of a latin square.
2. A permutation represented by two images is used to describe a pairwise row permutation. If $\mathbf{a} = \{a_1 a_2...a_n\}$ and $\mathbf{b} = \{b_1 b_2...b_n\}$ are images of two permutations then by $[\mathbf{a}, \mathbf{b}]$ we denote a permutation such that $p(a_k) = b_k$, $k = 1,...,n$.
3. A permutation represented by disjoint cycles is used to classify the pairwise row permutations. The set of cycle lengths $\{m_1,...,m_s\}$ we call the shape of the permutation and denote by $t(\mathbf{a}, \mathbf{b})$. Note that $\{m_i\}$ is a partition of n, thus $m_1+...+m_s = n$.

The permutation representations above are equivalent. Thus the first representation is a special case of the second representation with $\mathbf{a} = \mathbf{I} = \{1\ 2\ ...\ n\}$, the second can be easily transformed into the first by applying the inverse permutation $\mathbf{a}^{-1}$ to both vectors which reduces **a** to **I**, and the third is a different notation of the second representation.

The concept of *even* and *odd* permutation is defined for the image representation as follows: If $i_k > i_m$ for $k < m$ in $\{i_1\ i_2 ... i_n\}$, then i_k and i_m are said to be transposed. If the number of transpositions of $\{i_1\ i_2 ... i_n\}$ is even (odd) then the permutation itself is said to be even (odd).

Combinatorial propositions (stated without proof):

1. A cycle of length s is the product of s - 1 transpositions.
2. If the number of transpositions of a permutation is even (odd) then the permutation is even (odd).
3. If two permutations [**a**, **b**] and [**a**, **c**] have the same parity, then [**b**, **c**] is an even permutation, otherwise it is odd.

From propositions **1** and **2** we conclude that the parity of [**a**, **b**] coincides with the parity of $n - s$, where n is the degree of the permutation and s is the number of disjoint cycles of its shape. In particular, for even n the parity of the permutation coincides with the parity of the number of the cycles.

We establish a precedence order of permutations based on the shape and the lexicographic order. First, since the number of all possible shapes of the permutations of degree n is finite (= number of partitions of n), we enumerate them arbitrarily and thus establish a precedence order among the shapes. Thus, for our algorithm when $n = 8$, the order we use is as follows:

$$t_1 = \{2,6\},\ t_2 = \{3,5\},\ t_3 = \{4,4\},\ t_4 = \{2,2,2,2\},\ t_5 = \{8\},\ t_6 = \{2,2,4\},$$
$$t_7 = \{2,3,3\};\quad t_1 < t_2 < t_3 < t_4 < t_5 < t_6, \qquad (*)$$

where the numbers in the brackets indicate the lengths of the cycles of the pairwise permutations.

Second, if two permutations have the same shape we use the lexicographic order to determine the order:

Definitions. **A.** If $\mathbf{v}_1$ and $\mathbf{v}_2$ are distinct permutation images of order n then we say that $\mathbf{v}_1$ *precedes* $\mathbf{v}_2$ or $\mathbf{v}_1 < \mathbf{v}_2$ if

either 1. $t\,(\mathbf{I}, \mathbf{v}_1) < t\,(\mathbf{I}, \mathbf{v}_2)$,

or 2. $t\,(\mathbf{I}, \mathbf{v}_1) = t\,(\mathbf{I}, \mathbf{v}_2)$, and $\mathbf{v}_1$ lexicographically precedes $\mathbf{v}_2$.

Since rows of a latin square are represented by $\mathbf{v}$, the precedence order of latin squares can be established as follows:

B. If V and V' are distinct latin squares of order n then we say that V < V' if there exists a $k < n$ such that $\mathbf{v}_1 = \mathbf{v}_1'$,..., $\mathbf{v}_{k-1} = \mathbf{v}_{k-1}'$, and $\mathbf{v}_k < \mathbf{v}_k'$.

A permutation $\mathbf{v}$ of degree n we call a *candidate* for a row of a latin square of order n. Two candidates $\mathbf{v}_1$ and $\mathbf{v}_2$ are *compatible* if $[\mathbf{v}_1,\mathbf{v}_2]$ has no fixed points; i.e., the shape $t\,(\mathbf{v}_1,\mathbf{v}_2) = (m_1,\ldots,m_s)$ has each $m_i > 1$. If we have a latin rectangle of r rows, then a vector $\mathbf{v}$ compatible with all r rows of this rectangle is said to be a candidate for the $(r+1)$-th row. Obviously, the candidates for the $(r+1)$-th row are among the candidates for every previous row of the rectangle.

A transformation α is said to *lexically reduce* the latin rectangle V if $V' = \alpha(V)$ and $V' < V$. Since we are interested in the main classes, α can be any combination of **ColP, RowP, EntR, RowI** for latin rectangles $k \times 8$, where $1 \le k \le 8$. **MatT**, matrix transpose, is applicable only to complete latin squares.

Among all the members of a main class we choose as the class representative, or *certificate,* the latin square M such that in its main class there is no V < M ; we also refer to M as to the *minimal member* of a class .

Since a certificate is defined under a certain set of possible transformations, we can extend the certificate definition to the isotopy classes under **ColP, RowP, EntR** transformations as well as to the classes of latin rectangles under **ColP, RowP, EntR,** and **RowI** transformations.

Certificates have the following properties:

1. The first row of a certificate is the identity. This property follows from the fact that the relabelling of entries (it leaves the latin square in the same isotopy class) which transforms the first row of a latin square into the identity lexically reduces the latin rectangle itself.
2. The second row $\mathbf{v_2}$ of a certificate with the shape $t\,(\mathbf{v_1}, \mathbf{v_2})$, where $\mathbf{v_1}$ is its first row, has the minimal lexicographic order of any candidate of this shape. This property follows from the fact that no column permutation of a latin square changes the shape of the row permutations – it can only change the order of their cycles, which is irrelevant in our classification. Therefore, we may apply a column permutation to produce a form of $\mathbf{v_2}$ in which each cycle appears in consecutive indices and the cycles appear in non–decreasing order of length. E.g., (21 453 786) is the minimal form of permutation with cycle structure {2, 3, 3}. If we then relabel the entries to transform the first row into the identity, we, at the same time, obtain the minimal order of the second row. We leave this without proof, since it can be very easily done by induction . Thus, the shape $t\,(\mathbf{v_1}, \mathbf{v_2})$ uniquely defines the second row.
3. At least half of the rows of a certificate are even permutations with respect to each other.

   This follows from the combinatorial proposition **3** that a pairwise permutation of

any two permutations of the same parity is even. Following (*), these rows must be the first rows of a latin square.

4. Any k-row rectangle (first $k < r$ rows) of a certificate is a certificate itself under **ColP, RowP, EntR, RowI** transformations. Otherwise a transformation which lexically reduces the rectangle would lexically reduces the V itself. As a result, one can conclude that if v_i and v_j are two chosen candidates for i-th and j-th rows, then $v_i < v_j$. In particular, the shapes $t_{12}, t_{13}, \ldots, t_{1r}$ are in a non-decreasing order, i.e. $t_{12} \leq \ldots \leq t_{1r}$, where $t_{1i} = t(v_1, v_i)$.

Based on the above properties, the main points of the algorithm can be stated as follows:

A. Following property 1, the first row of any solution must be the identitity

$$\mathbf{I} = 12345678.$$

B. For $n = 8$, there are seven shapes which do not fix any element of **I**. They are:

$$\{2,6\},\ \{3,5\},\ \{4,4\},\ \{2,2,2,2\},\ \{8\},\ \{2,2,4\},\ \{2,3,3\}.$$

The first four shapes have an even numbers of cycles and so the permutations of these shapes are even as shown above. We order them sequentially.

C. Following property 3 (the second row is even) and property 2, there are exactly four candidates for the second row:

$$21456783,\quad 23156784,\quad 23416785,\quad \text{or}\quad 21436587,$$

one of each of the four even shapes. So the backtracking algorithm can be applied starting with the four initial latin rectangles:

$$L_1 = \begin{matrix}12345678\\21456783\end{matrix} \qquad L_2 = \begin{matrix}12345678\\23156784\end{matrix}$$

$$L_3 = \begin{matrix}12345678\\23416785\end{matrix} \qquad L_4 = \begin{matrix}12345678\\21436587\end{matrix}$$

D. To save time, the first row constructed by the algorithm is v_3, the third row of the latin square. We initialize four different sets D_3', D_3'', D_3''', and D_3'''' of all possible candidates compatible with L_1, ..., L_4 : one set for each initial latin rectangle. Following property 4, the candidates for D_3' have their shapes among

$$T = \{(2,6),\ (3,5),\ (4,4),\ (2,2,2,2),\ (8),\ (2,2,4),\ (2,3,3)\},$$

the shapes of candidates of D_3'' lie in $T \setminus (2,6)$, of D_3''' lie in $T \setminus \{(2,6),(3,5)\}$, and so on.

D_3', D_3'', D_3''' and D_3'''' are computed by a separate program. The order of the candidates in each set is the precedence order defined earlier.

By definition of D_i, $D_i \supset D_{i+1}$, or, if v is a chosen candidate for the i-th row, any $v' > v$ which is in D_i and compatible with v must belong to D_{i+1}. The maintenance of the domains D_i, required by the backtracking method, is based on this single condition.

As mentioned in Chapter 2, the transformation algorithm is applied in this part to reduce the search tree. The symmetry group passed to the transformation algorithm differs depending on the order of the rectangle: it is $S_2 \times S_8 \times S_k$; row inverse permutation is mapped onto S_2, a column permutation onto S_8, and a row permutation of a $k \times 8$ rectangle onto S_k. When a latin square is completed, the symmetry group is $S_3 \times S_8 \times S_8$, where the conjugacy operations are mapped onto S_3.

Let us now make some comments on the interface of the generation and the transformation algorithms.

Following property 4, if a chosen row gives rise to a non–certificate rectangle under **ColP, RowP, EntR, RowI,** then this rectangle need not be extended, since it cannot give rise to any certificate of a higher order. To test whether a latin rectangle is a certificate, the transformation algorithm is applied to it. Because this algorithm is time consuming, the strategy of its application is as follows: the algorithm is applied to 3×8, 4×8, and 5×8 latin rectangles; after that every rectangle that survives is completed to latin squares in all possible ways; then the algorithm is applied to the 8×8 latin squares to test

each of them for being a certificate under all the isomorphism operations **ColP, RowP, EntR, RowI, MatT.**

At the intermediate levels of the search, when the transformation algorithm is applied, the number of certificates is counted using three types of classification: under the isotopy transformations (**ColP, RowP, EntR**), under **ColP, RowP, EntR, RowI** (including the row inverse permutation), and by the number of even and odd pairwise row permutations.

To each 8×8 latin square built, the conjugacy operations are applied (a latin square is, first, considered as an 8×8 rectangle, and the inverse operation is applied to it; when the computation of the latin square is completed, all remaining conjugacy operations are applied to it) The last step of this part is to test whether an initially constructed latin square (a certificate under **ColP, RowP, EntR, RowI**) is the certificate in the main class (under **ColP, RowP, EntR, RowI, MatT**). At the same time, a consistency check (this check is done under conjugacy operations) is performed. Whenever the certificate of a main class is found, it is stored along with the order of its automorphism group under the conjugacy and the isotopy operations.

3.3. Results of the Latin Square Generation.

Following the Dénes classifications [4], the results obtained for the 8×8 latin squares are classified by the main classes as well as by the isotopy classes. The results are given below, along with the order of the automorphism groups of the representatives.

The distribution of the main classes by the order of the automorphism group of their representatives is given in Table 1, where N_j^m denotes the number of main classes whose representatives have automorphism of order $|Aut_j^m|$.

$\lvert Aut_j^m \rvert$	N_j^m	$\lvert Aut_j^m \rvert$	N_j^m	$\lvert Aut_j^m \rvert$	N_j^m
1	270 611	18	9	126	1
2	11 119	20	2	128	4
3	213	21	1	192	5
4	1089	24	28	256	4
5	1	32	34	288	1
6	158	36	2	384	4
8	227	48	8	576	2
9	1	64	11	1536	3
10	3	72	1	3072	2
12	35	84	1	9216	1
16	69	96	6	64512	1
The total number of the main classes is					283657

Table 1. Distribution of main classes of 8×8 latin squares

by the order of automorphism groups.

The distribution of the isotopy classes by the order of the automorphism group of their representatives is given in Table 2, where N_j^i denotes the number of isotopy classes whose representatives have automorphism groups of the order $\lvert Aut_j^i \rvert$.

$\lvert Aut_j^i \rvert$	N_j^i	$\lvert Aut_j^i \rvert$	N_j^i	$\lvert Aut_j^i \rvert$	N_j^i
1	1644434	10	12	96	8
2	28 767	12	55	128	12
3	310	16	172	192	6
4	1 854	24	21	256	3
5	12	32	36	512	2
6	136	42	5	1536	1
7	2	48	7	10752	1
8	397	64	14		
The total number of the isotopy classes is					1676267

Table 2. Distribution of isotopy classes of 8×8 latin squares

by the order of automorphism groups.

3.4. Analysis of the Results.

Having found the number of main and isotopy classes and also the order of the automorphism group of their members, the total number of reduced latin squares can be computed using Tab. 1 or Tab. 2. This result is of interest since it can be compared with Wells' computation of the number of 8×8 reduced latin squares .

Each isotopy class contains $(n!)^3$ latin squares. Let us denote the number of reduced squares of an isotopy class by $N^{(ri)}$. Then

$$N^{(ri)} = n \times (n!).$$

In fact, apply column permutations ($n!$ operations), then choose an arbitrary first row (n operations); the permutation of the other rows is forced (since the order of the entries in the first column and row must be the same) as well as the relabelling of entries in order to reduce the first row and column to the identities.

Since $| Aut_j^i |$ is the automorphism group of every element of the same isotopy class, the number of reduced latin squares in one isotopy class is equal to

$$\frac{N^{(ri)}}{| Aut_j^i |},$$

and the total number of reduced latin squares is equal to

$$N^{(r)} = N^{(ri)} \sum_j \frac{N_j^i}{| Aut_j^i |},$$

where $| Aut_j^i |$ is the order of the automorphism group of a representative of an isotopy class.

The same consideration may be applied to the main classes:

The number of reduced latin squares of a main class is

$$N^{(rm)} = 3! \times n \times (n!)$$

$$N^{(r)} = N^{(rm)} \sum_j \frac{N_j^m}{| Aut_j^m |}.$$

The results of Table 1, as well as those of Table 2, give the same number of reduced latin squares of order 8. It is

535 281 401 856,

and coincides with Wells' result [18].

As mentioned, the following results on enumeration of the 8×8 latin squares are known:

The number of reduced latin squares obtained by Wells is

$$N^{(r)} = 535\ 281\ 401\ 856.$$

The number of main classes obtained by Arlazarov is

$$N^{(m)} = 283\ 640.$$

The number of isotopy classes obtained by Brown is

$$N^{(i)} = 1\,676\,257.$$

Our results coincide with those of Wells, while showing discrepancies with those of Arlazarov and Brown.

The number of main classes obtained by Arlazarov *et al.* is lower by 17 than in Table 1.

Since Arlazarov *et al.* tabulate the order of the automorphism groups, we are able to show that their results do not agree with those of Wells.

To compare our results with the Brown's, who gives the distribution using the classification by number of even and odd pairwise row permutations, we have collected the same information during the generation process, and show the places where the discrepancies occur.

The distribution of isotopy classes using Brown's classification is given in Table 3, where the pair (i, j) indicates the number of even and odd rows respectively. For complete squares, $i + j = 8$. As one can see, the discrepancies occur in two results: for (6,2) and (7,1), i. e. in the number of latin squares which have six even rows, and in the number of latin squares which have seven even rows.

Latin Rectangles	# of Rectangles	Remarks
(2, 0)	4	
(3, 0)	110	
(4, 0)	11 968	
(5, 0)	296 227	
(4, 1)	1 478 186	no Brown's result

Latin Squares	#of Latin Squares	
(4, 4)	466 867	
(5, 3)	720 840	
(6, 2)	372 350	Brown's result is 372 344
(7, 1)	101 786	Brown's result is 101 782
(8, 0)	14 424	
Total	1 676 267	Brown's result is 1 676 257

Table 3. Distribution of k×8 latin rectangles by number of even and odd permutations (compare with [3]).

CHAPTER 4. Extension of a Partial Incidence Matrix.

Part I of the algorithm creates the list of 8×8 latin squares, each of which can be mapped onto the first 27 columns of an incidence matrix in the standard form given in Fig. 3. In this chapter, we describe Part II of the algorithm: first, the extension of the first 27 columns into 40 columns, and then the extension of 40 columns into complete incidence matrices.

4.1. Standard Form of 40 Columns.

A *candidate* for column A^j (j = 28,..., 40) is a binary vector of 91 elements with the following properties:

A^j can be a column of the incidence matrix, i. e., it contains 10 ones and 81 zeros;

A^j has a *standard form* of the j-th column; by the standard form we mean a vector with two fixed ones at specific places (specific for the j-th column; this form is given below in Fig. 6); and

A^j is compatible with the given 27 first columns, i. e., the inner product of A^j with each of the 27 given columns is equal 1.

Two candidates A^i, A^k (i, k = 28,..., 40) are said to be *compatible* if they are candidates for different columns ($i \neq k$) and if the inner product of A^i and A^k is equal 1.

A *standard form* of the first 40 columns is shown in Fig. 6. As one can see, the first 27 columns coincide with the ones of Fig. 3. The column block $A^{28:34}$ contains 7 columns, and $A^{35:40}$ – 6 columns. The structure of blocks C^1, C^2 and D^2 is constant (see Fig. 7), i. e., their entries do not depend on the structure of the first 27 columns; the

structure of block D^1 has three different representations (one of them is shown in Fig. 7) depending on the structure of block B^2.

We state the following proposition:

Proposition 4.1. Any incidence matrix can be transformed into a form such that its first 40 columns are in the standard form.

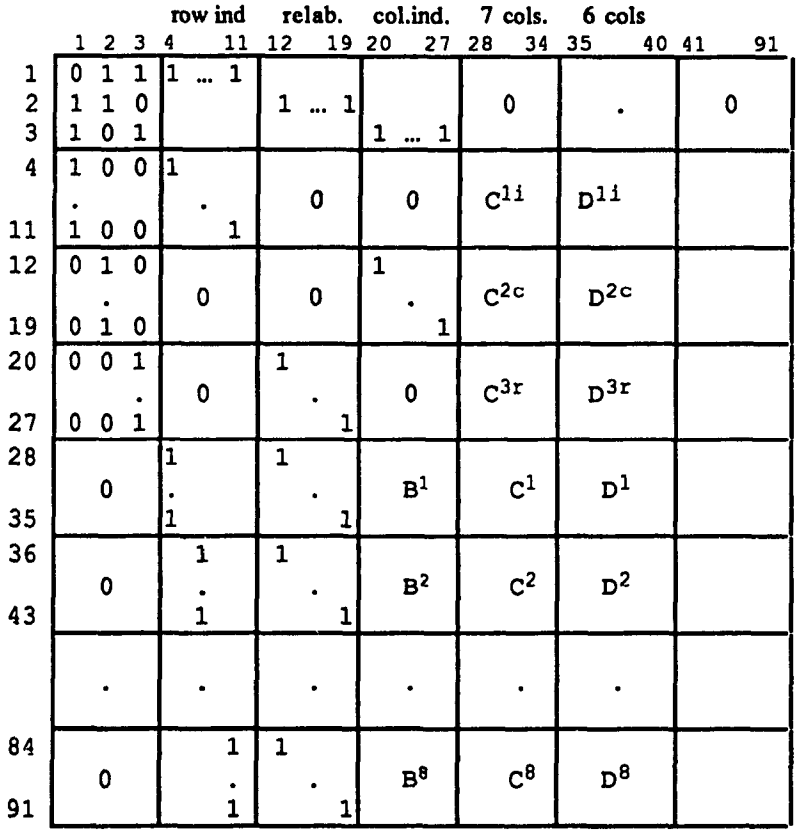

Fig. 6. Standard form of 40 columns

The blocks C^{1i}, D^{1i} of Fig. 6 are annotated '1*i* ' to show that any column of this block intersects with two previous columns: the first column A^1 and one of the columns of the row index block $A^{4:11}$. Notations C^{2c}, D^{2c}, and C^{3r}, C^{3r} have a similar meaning (*c* stands for column index block, and *r* stands for the relabelling block).

Proof. First we give an informal proof, or rather a sketch of the algorithm how an incidence matrix can be reduced to the form required.

Since B^1,..., B^8 are mapped onto a latin square ($B^1 \rightarrow v_1$, ..., $B^8 \rightarrow v_8$), whose first row (v_1) is the identity and the second row (v_2) has 2 as its first entry and 1 or 3 as its second entry, we may force the ones of rows A_{28} (block C^1) and A_{38} (block D^2). Then by permuting the columns inside the blocks $A^{28:34}$ and $A^{35:40}$, we may force one more 1 of each of these columns.

	28	34	35	40
28	C^1	1111111	D^1	000000
29		0000000		000000
30		0000000		000000
31		0000000		100000
32		0000000		010000
33		0000000		001000
34		0000000		000100
35		0000000		000010
36	C^2	0000000	D^2	000000
37		0000000		000000
38		1000000		111111
39		0100000		000000
40		0010000		000000
41		0001000		000000
42		0000100		000000
43		0000010		000000

(D^1 is an example for the second row of Latin square 23156784 or 23416785)

Fig. 7. Structure of blocks C^1, C^2, D^2 and an example of D^1.

To prove the proposition, let us be more precise and recall some points of previous chapters on the correspondence between a latin square and the column block $A^{20:27}$.

If $V = \{v_{kj}\}$ is a latin square, where $k, j = 1,..., 8$, and $v_{kj} = i$, then $b^k_{ij} = 1$ and $b^k_{i'j} = 0$ for $i' \neq i$. This is equivalent to:

- the k-th row of a latin square is mapped into block B^k, $k = 1,..., 8$; and

- the (k, j)-th entry of the latin square determines the position of the only 1 in the j-th column.

The correspondence between the block elements $B^k = \{b_{ij}^k\}$ and entries of the incidence matrix A is

$$a_{27+8(k-1)+i,\ 19+j} = b_{ij}^k .$$

By construction, the first row of any latin square generated is

12345678

and the second row is one of the following four:

21456783, 23156784, 23416785, or 21436587. (*)

Thus,

$b_{i\,i}^1 = a_{27+i,\ 19+i} = 1,$ ($*_1$)

$b_{2\,1}^2 = a_{37,\ 20} = 1,$ ($*_2$)

$b_{x\,2}^2 = a_{35+x,\ 21} = 1$, where x = 1 or 3. ($*_3$)

Let A be any incidence matrix with the first 27 columns given in the standard form. The fact that columns A^{28},..., A^{91} can be permuted in such a way that blocks C^1, C^2, and D^2 have the structures of Fig. 7 follows from the following properties of the projective planes :

First we rearrange the columns A^{28},..., A^{91} to reduce A^{28},..., A^{34} to the standard form:

C1. Row A_{28} has three ones in the first 27 columns (columns A^4, A^{12}, A^{20}); therefore, there are seven more columns with 1 in row A_{28}. Let us move them into column block $A^{28:34}$.

C2. No more ones can be in block C^1, because of intersection of its columns with column A^4.

The correctness of structure C^1 (see Fig. 7) follows from C1, C2.

C3. Since the ones in the columns of block C^1 (row A^{28}) determine their intersection with A^{12}, these ones induce zeros in row A_{36} of C^2.

C4. ($*_2$), another fixed 1 of column A^{20}, induces that row A_{37} of block C^2 must contain only zeros.

C5. The columns A^{28}, ..., A^{34} can intersect A^5 only in block C^2 or in row A_5, therefore, since we have 7 columns to intersect A^5 and exactly 7 rows where it may happen, we permute A^{28}, ..., A^{34} to obtain the required structure of C^2. At the same time we fix $a_{5,34} = c_{2\ 7}^{1i} = 1$.

The correctness of structure C^2 (see Fig. 7) follows from C3 – C5.

Now we rearrange the columns A^{35},..., A^{91} to reduce A^{35},..., A^{40} to the standard form:

D1. The row A_{38} has four ones in the first 34 columns; therefore, there are six more columns with the 1 in row A_{38}. Let us move them by permuting the columns A^{35},...,A^{91} in the column block $A^{34:40}$.

D2. No more ones can be in block D^2 because of intersection of its columns with column A^5.

The correctness of structure D^2 (see Fig. 7) follows from D1, D2.

D3. Since the ones in the columns of block D^2 (row A^{38}) determine their intersection with A^{28}, these ones induce zeros in row A_{28} of C^2.

D4. Since the columns A^{35},..., A^{40} intersect A^{14}, the ones on row A^{38} induce zeros on row A_{30} in block D1.

D5. ($*_3$), case x = 3 : $a_{38,21} = 1$, and ($*_1$): $a_{29,21} = 1$ determine the intersection of column A^{21} with all the columns of block $A^{35:40}$ in row A_{38}, thus the row A_{29} in block D^2 must contain only zeros. (similar conclusion about another row of D^1 can be done for x = 1).

D6. The columns A^{35}, ..., A^{40} can intersect A^4 only in block D^1 or in row A_4, therefore, since we have 6 columns to intersect A^4 and exactly 6 rows where it may happen, we permute A^{35}, ..., A^{40} to obtain the required structure of D^1. At the same time we fix $a_{4,40} = d_{1\,6}^{1\times i} = 1$.

The correctness of structure D^1 follows from D3 – D6.

Thus, an incidence matrix can be transformed into a form of Fig. 6 with the blocks C^1, C^2 D^1, D^2 of Fig. 7 (plus $a_{4,40} = a_{5,34} = 1$). Note that the columns A^{28},..., A^{40} have two ones fixed.

4.2. Description of the Algorithm.

Part IIa of the algorithm implements the backtracking method with the use of the compatibility matrix (for an outline of this algorithm, see Chapter 2). To construct the compatibility matrix, the candidates for the columns A^{28} to A^{40} must be computed first. This part of the algorithm is adapted to the problem by exploiting the particular properties of the standard form.

Two ones in each column are fixed (since we consider them in the standard form), therefore, there are eight more ones to be placed in each column.

Fig. 6 shows that each column (A^{28},...,A^{40}) is divided into the following nine blocks:

C^3,..., C^8, C^{1i}, C^{2c}, C^{3r}, (columns A^{28},..., A^{34}), or

D^3,..., D^8, D^{1i}, D^{2c}, D^{3r} (columns A^{35},..., A^{40}).

The distribution of ones inside the above blocks has the following properties, determined by the standard form representation:

1. The blocks C^i (similarly, D^i) ($A^{28:34}_{27+8(i-1)+1:27+8i}$, where i = 3,..., 8) do not contain more than one 1 in each column and in each row. This property follows from the following consideration:

- Column A^i has 8 ones in the i-th row block. Therefore, no column of C^i (D^i) can have more than one 1;

- Row A_{28} (A_{38} for rows of D^i) have 7 (6 for D^i) ones. Therefore no row of C^i (D^i) can have more than one 1.

2. The intersection of the columns A^{28} through A^{40} with columns A^1, A^2, and A^3 can be only in blocks C^{1i}, C^{2c}, C^{3r}(D^{1i}, D^{2c}, D^{3r}), i. e. each column must have exactly one 1 in these blocks.

3. If c^k_{ij} (similarly, d^k_{ij}) = 1, then $c^{k'}_{ij} = 0$ for any k' such that $k' \neq k$ (because of intersection with A^{11+i}). For each A^j if there are k, i such that $c^k_{ij} = 1$, then $c^{1i}_{ij} = 0$ (because of intersection with A^{3+k}), and $c^{3r}_{ij} = 0$ (because of intersection with A^{11+i}). Similarly, for each A^j, if there are k, s such that $b^k_{sj} = 1$ then $c^{2c}_{sj} = 0$ (because of intersection with A^{19+j}).

Based on the above properties, the algorithm for the generation of the candidates is as follows:

For each A^j, j = 28 to 34, we choose consecutively t = 3,..., 6 and assume that C^t contains no ones in column A^j. For u = 3,...,8, $u \neq t$ we compute the set X_u of all possible i such that $C^u_{ij} = 1$, as follows:

$$X_u = (\{1,...,8\} \setminus \{i_m \mid c^m_{i_m j} = 1, m = 1,...,u-1\}) \setminus$$

$$\{i_u \mid c^m_{i_m j} = 1, \text{there is a } s, \text{ such that } b^m_{i_m s} = b^u_{i_u s} = 1, m = 1,...,u-1\}.$$

When five blocks out of six are completed for A^j, the remaining blocks C^{1i}, C^{2c}, C^{3r} are computed as follows:

. $c^{1i}_{tj} = 1$ – to intersect A^{3+t}, where t is such that C^t contains no ones in A^j.

$c^{3r}_{sj} = 1$, where $s \in \{1,..., 8\} - \{i \mid c^{u}_{ij} = 1, u = 1,..., 8\}$;

$c^{2c}_{sj} = 1$, where $s \in \{1,..., 8\} - \{s \mid c^{u}_{ij} = b^{u}_{is} = 1, u = 1,..., 8\}$.

A similar computation is applied to columns $A^{35},..., A^{40}$.

The above information is sufficient to understand the algorithm which generates the candidates for columns $A^{28},..., A^{40}$. Each of the computed candidates is stored as an array of bit vectors (of 91 bits = 12 bytes). To build the compatibility matrix, the inner product of each pair of candidates for different columns is computed using the bitwise AND operation. The entries of the compatibility matrix are assigned 1 if the inner product is 1, and 0 otherwise. The rows of the *LIVE* matrix are computed in advance up to the column A^{34}. This allows us to perform the "lookahead" till A^{34}. For candidates (columns $A^{35},...,$ A^{40}), the appropriate rows of *LIVE* are computed up to the next column, i. e., there is no "lookahead" strategy starting from A^{35}. This strategy was adopted based on the shape of the search tree.

4.3. Result of Extension to 40–column Partial Incidence Matrices (Part IIa).

The above algorithm was applied to all 283657 non-isomorphic 8×8 latin squares previously obtained (see Chapter 3). Only 21 of them can be extended to 40 columns. They give rise to **326 40–column partial incidence matrices.** Table 4 gives the order of the isotopy classes for each latin square as well as the order of the group under the conjugacy operations. The latin squares themselves can be found in Appendix B.

LS id	Automorphism Conj.	Isot.	LS id	Automorphism Conj.	Isot.	LS id	Automorphism Conj.	Isot.
1	6	1	8	2	6	15	1	12
2	6	1	9	6	3	16	2	48
3	2	3	10	1	2	17	2	192
4	2	6	11	2	3	18	2	6
5	2	2	12	6	6	19	2	128
6	2	6	13	6	1	20	6	1536
7	1	1	14	2	2	21	6	256

Table 4. The order of the automorphism groups of 21 latin squares which extend to 40–column incidence matrices.

4.4. Result of Extension to Complete Incidence Matrices (Part IIb).

Part IIb of the algorithm extends each of the 326 partial matrices which were obtained in Part IIa into complete incidence matrices using the "PLANE91" program written by Thiel. First, a small program ("MAKE_LARRY_INPUT") transforms the compact output form of an incomplete matrix into the form required by "PLANE91". Then the program PLANE91 is applied. The result of Part IIb is that **325 matrices out of 326** have extensions to complete planes. Each of these 325 matrices has exactly one extension.

Appendix C shows the 40–column partial matrix which has no extensions to a complete matrix.

Chapter 5. Plane Identification

In this chapter we discuss an algorithm for plane identification. It is achieved by transforming incidence matrices into a *canonical* form, which is unique for any given matrix. A straightforward comparison of the matrices in the canonical form allows to determine the non–isomorphic classes of the incidence matrices.

5.1. Canonical Form of an Incidence Matrix.

We define a standard form for complete incidence matrices as we did for incomplete matrices. An incidence matrix of a plane is said to be in a *standard form* if its first 19 rows and 19 columns have the form of Fig. 8. In contrast to the standard form of 40 columns, each column block, $A^{4:11}$ to $A^{84:91}$ has 8 columns.

An incidence matrix C is said to be a *plane certificate* , or a *canonical form* of a plane, if C has the following properties:

1. the first 19 rows and 19 columns of C have the form of Fig. 8;
2. if A is another incidence matrix of the plane with property 1, then there are s , t such that for $j < t$, $a_{ij} = c_{ij}$, for $i = 20,...,91$, $a_{it} = c_{it}$, , for $i = 20,..., s - 1$, and $c_{st} < a_{st}$.

Let us define an *order* among any two distinct incidence matrices (A and B) of the same plane as follows:

A *precedes* B if either A is in the standard form (Fig.8), and B is not;

or A and B are both in the standard form, and there are s , t such that for $j < t$ $a_{ij} = b_{ij}$, for $i = 20,...,91$, $a_{it} = b_{it}$, for $i = 20,..., s - 1$, and $a_{st} < b_{st}$.

A transformation which reduces B to A is said to lexically reduce B (make it 'better'), a transformation which reduces A to B is said to lexically raise A (make it

'worse'), and a transformation which transforms a matrix into itself is said to be automc.phic ('stable').

Proposition 5.1. Any incidence matrix can be transformed into the standard form of a complete incidence matrix.

Proof. We have shown in Chapter 1 that, by permutation of rows and columns, we can transform the first 19 columns into the standard form required. The structure of blocks $A^{28:35}$, $A^{36:43}$,..., $A^{84:91}$ can be obtained, first, by moving the ones of row block $A_{4:11}$ (as shown on Fig. 8) and, then, by permuting independently the columns of the above blocks.

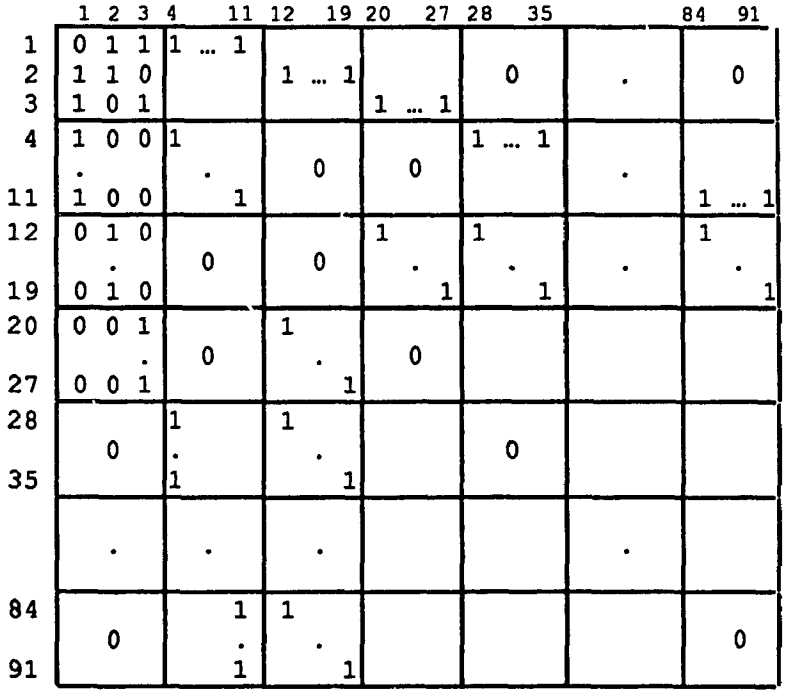

Fig. 8. Standard form of a complete incidence matrix.

Part III of the algorithm transforms a complete incidence matrix into its certificate. Since the representation of a plane by the certificate is unique, two planes can be compared by comparing their certificates.

To find the certificates of the 325 complete incidence matrices, we applied the transformation algorithm (see Chapter 2).

The symmetry group for the algorithm is defined as S_{91} to map the column permutations (the row permutations are forced by the standard form of Fig. 8).

5.2. Results of Plane Identification.

Part III of the algorithm identifies the certificates of the 325 planes obtained. For this purpose the certificates of the four known planes are computed and each of the 325 certificates is compared with the four known ones.

The result of the comparison is that the 325 planes are distributed among the four known non–isomorphic classes of planes: Desarguesian, Hughes, and right and left near–field planes. The distribution of the planes is summarized in Table 5. The latin squares themselves appear in Appendix B.

LS id	automorphism Conj.	Isotopy	Right	Left	Hughes	Desarguesian
1	6	1	1			
2	6	1		1		
3	2	3			1	
4	2	6			4	
5	2	2			1	
6	2	6	1	1		
7	1	1			1	
8	6	1			1	
9	6	3			1	
10	1	2			1	
11	2	3			1	
12	6	6	3	3		
13	6	1			1	
14	2	2			1	
15	1	12			1	
16	2	48	1	1		
17	2	192			8	
18	2	6			1	
19	2	128			8	
20	6	1536	108	108	64	
21	6	256				2
Total number			114	114	95	2

Table 5. Distribution of 325 complete incidence matrices.

CONCLUSION. Analysis of the Results.

It is not clear how to check the results obtained. Some internal consistency checking can be done (Lam, Kolesova [13]), which does not prove the correctness of our results but at least shows the absence of certain contradictions. The consistency checking proposed by Lam is based on one–to–one mapping between a latin square and a choice of the first three columns of a plane:

Since a choice of the first three non–collinear columns (a *triangle*) of a plane determines a latin square (see Chapter 1), the following must be satisfied:

(1a) All latin squares which can be obtained from the four known planes (by different choice of rectangles) must be in the list of our results given in Table 5.

(1b) Moreover, all the latin squares which can be 'cut' from a plane must give rise to the plane in our results.

(2) Suppose we choose a triangle of a plane. Let $|Aut_3|$ be the order of the group of plane transformations which fixes the triangle. A latin square corresponds to the triangle. We denote the order of its automorphism group by $|Aut_{ls}|$. Since each transformation of the latin square corresponds to one of the plane (which fixes the triangle) we may compute the number of 'different' copies of the plane with the given triangle. It must be

$$\frac{|Aut_{ls}|}{|Aut_3|},$$

'Different' in this context means non–automorphic copies of the plane which have the same triangle. The number of different (expected) copies corresponding to the latin squares must coincide with the number of copies in Table 5.

The computations necessary for the above consistency checking have been made for the four known place. $|Aut_3|$ were computed using the generators of the automorphism group obtained by transformation algorithm. To show that results on $|Aut_3|$ are consistent, the correctness of the following formulae can be checked for each plane:

$$|Aut_{plane}|(\frac{1}{c_1} + ... + \frac{1}{c_k}) = \frac{91 \times 90 \times 81}{6},$$

where $|Aut_{plane}|$ is the order of the automorphism group of a plane, $\frac{91 \times 90 \times 81}{6}$ is the number of non–collinear columns of a plane, and $c_1, ..., c_k$ are $|Aut_3|$, the orders of the automorphism groups of all different triangles. For instance, for the right (left) near-field plane we have

$$311\ 040 \times (\frac{1}{6} + \frac{1}{12} + \frac{1}{12} + \frac{1}{96} + \frac{1}{768}) = \frac{91 \times 90 \times 81}{6},$$

where 311 040 is the order of the isomorphism group of the left (right) near–field plane, and 6, 12, 12, 96, and 768 are the orders of the automorphism groups which fix its 3-sets (Table 7).

The summary of the check computations are given in Tables 6–8. These tables show that the list of all latin squares obtained from the four planes coincides with the list given in Table 5 and, moreover, the latin squares associate with the correct planes. The number of distinct copies associated with each latin square given in Table 5 agrees with the expected copies obtained by the check computations.

Triangle	L.S. id	Automorphism Triangle	Automorphism L.S.	Expected Copies
1	21	768	1536	2

Table 6. The latin square and the triangle

relating to the Desarguesian plane

Triangle	L.S. id	Automorphism Triangle	Automorphism L.S.	Expected Copies
1	1 (2†)	6	6	1
2	12	12	36	3
3	6	12	12	1
4	16	96	96	1
5	20	96	9216	96
6	20	768	9216	12

Table 7. The latin squares and the triangles

relating to the right (†left) near–field plane

Triangle	L.S. id	Automorphism Triangle	Automorphism L.S.	Expected Copies
1	4	12	12	1
2	15	2	12	1
3	10	2	2	1
4	5	4	4	1
5	11	6	6	1
6	19	32	256	8
7	17	48	384	8
8	18	12	12	1
9	14	4	4	1
10	3	6	6	1
11	13	6	6	1
12	7	1	1	1
13	4	4	12	3
14	8	6	6	1
15	9	18	18	1
16	20	144	9216	64

Table 8. The latin squares and the triangles

relating to the Hughes plane

REFERENCES

[1] В. Л. Арлазаров, А. М. Бараев, Я. Ю. Гольфанд, И.А. Фараджев, "Построение с помощью ЭВМ всех латинских квадратов порядка 8", *Алгоритмические исследования в комбинаторике*, Москва, Наука, 129-140, (1978).

[2] S. E. Bammel, J. Rothstein, "The number of 9 x 9 Latin Squares", *Discrete Mathematics* , **11**, 93 - 95, (1975).

[3] J. W. Brown, "Enumeration of Latin Squares with Application to Order 8", *Journal of Combinatorial Theory* , **5**, 177-184, (1968).

[4] J. Dénes, A. D. Keedwell, *Latin Squares and their Applications* , Academic Press, (1978).

[5] M. Hall Jr. "Some Existence Problems for Designs", *Congressus Numerantium* , **50**, 111 - 128, (1985).

[6] M. Hall Jr., *The Theory of Groups* , MacMillan, 346 - 420 (1959).

[7] M. Hall Jr., J. Dean Swift and R. Killgrove, "On Projective Planes of Order Nine", *Mathematical Tables and Other Aids to Computation* , **XIII**, №. 68, October, (1959).

[8] D. R. Hughes, F. C. Piper, *Projective Planes* , Springer-Verlag, (1973).

[9] R. B. Killgrove, "A note on the Nonexistence of Certain Projective Planes of Order Nine", *Mathematical Tables and Other Aids to Computation* , **XIV**, №. 69, (1960).

[10] R. B. Killgrove, E. T. Parker, "Low order Projective Planes", *Proceedings of the 7-th Southeastern Conference on Combinatorics* , Graph Theory and Computing, 365-390 (1976).

[11] G. Kolesova, C. W. H. Lam, L. Thiel, "On the Number of 8×8 Latin Squares", to appear in *Journal of Combinatorial Theory*, Series A.

[12] C. W. H. Lam, G. Kolesova, L. Thiel, "A Computer Search for Finite Projective Planes of Order 9", in preparation.

[13] C. W. H. Lam, L. Thiel, "Backtrack Search with Isomorph Rejection and Consistency Check", to appear in *Journal of Symbolic Computations* .

[14] C. W. H. Lam, L. H. Thiel, S. Swiercz, "A Computer Search for a Projective Plane of 10", in *Algebraic, Extremal and Metric Combinatorics* 1986, London Mathematical Society, Lecture Notes Series 131, ed Deza, Frankl and Rosenberg, 155-165, (1988).

[15] E. T. Parker, R. B. Killgrove, E. Milne, "A Note on Projective Planes of Order Nine", *Math. of Comp.* Vol. **18**,506-508, (1964).

[16] F. W. Stevenson, *Projective Planes* , W. H. Freeman and Company, (1972).

[17] L. H. Thiel, C. W. H. Lam, S. Swiercz, "Using a CRAY-1 to perform backtrack search", *Proc. of Second International Conference on Supercomputing* , San Francisco, **III**, 92-99, (1987).

[18] M. B. Wells, "The Number of Latin Squares of Order Eight", *Journal of Combinatorial Theory* , **3**, 98 - 99, (1967).

APPENDICES

All the following appendices are the computer listings of intermediate and final results.

One remark concerning the notation:

The intermidiate results, such as the standard forms of partial incidence matrices have three types of entries: 0, 1, and ".".

0 and 1 are the fixed values of the incidence matrices, and

"." is not yet fixed.

APPENDIX A1. The First 27 columns of an incidence matrix in the standard form

```
                 11 11111111 22222222
    123 45678901 23456789 01234567

  1 011 11111111 00000000 00000000
  2 110 00000000 11111111 00000000
  3 101 00000000 00000000 11111111

  4 100 10000000 00000000 00000000
  5 100 01000000 00000000 00000000
  6 100 00100000 00000000 00000000
  7 100 00010000 00000000 00000000
  8 100 00001000 00000000 00000000
  9 100 00000100 00000000 00000000
 10 100 00000010 00000000 00000000
 11 100 00000001 00000000 00000000

 12 010 00000000 00000000 10000000
 13 010 00000000 00000000 01000000
 14 010 00000000 00000000 00100000
 15 010 00000000 00000000 00010000
 16 010 00000000 00000000 00001000
 17 010 00000000 00000000 00000100
 18 010 00000000 00000000 00000010
 19 010 00000000 00000000 00000001

 20 001 00000000 10000000 00000000
 21 001 00000000 01000000 00000000
 22 001 00000000 00100000 00000000
 23 001 00000000 00010000 00000000
 24 001 00000000 00001000 00000000
 25 001 00000000 00000100 00000000
 26 001 00000000 00000010 00000000
 27 001 00000000 00000001 00000000

 28 000 10000000 10000000 ........
 29 000 10000000 01000000 ........
 30 000 10000000 00100000 ........
 31 000 10000000 00010000 ........
 32 000 10000000 00001000 ........
 33 000 10000000 00000100 ........
 34 000 10000000 00000010 ........
 35 000 10000000 00000001 ........

 36 000 01000000 10000000 ........
 37 000 01000000 01000000 ........
 38 000 01000000 00100000 ........
 39 000 01000000 00010000 ........
 40 000 01000000 00001000 ........
 41 000 01000000 00000100 ........
 42 000 01000000 00000010 ........
 43 000 01000000 00000001 ........

 44 000 00100000 10000000 ........
 45 000 00100000 01000000 ........
 46 000 00100000 00100000 ........
 47 000 00100000 00010000 ........
 48 000 00100000 00001000 ........
 49 000 00100000 00000100 ........
 50 000 00100000 00000010 ........
 51 000 00100000 00000001 ........

 52 000 00010000 10000000 ........
 53 000 00010000 01000000 ........
 54 000 00010000 00100000 ........
 55 000 00010000 00010000 ........
 56 000 00010000 00001000 ........
 57 000 00010000 00000100 ........
 58 000 00010000 00000010 ........
 59 000 00010000 00000001 ........

 60 000 00001000 10000000 ........
 61 000 00001000 01000000 ........
 62 000 00001000 00100000 ........
 63 000 00001000 00010000 ........
 64 000 00001000 00001000 ........
 65 000 00001000 00000100 ........
 66 000 00001000 00000010 ........
 67 000 00001000 00000001 ........

 68 000 00000100 10000000 ........
 69 000 00000100 01000000 ........
 70 000 00000100 00100000 ........
 71 000 00000100 00010000 ........
 72 000 00000100 00001000 ........
 73 000 00000100 00000100 ........
 74 000 00000100 00000010 ........
 75 000 00000100 00000001 ........

 76 000 00000010 10000000 ........
 77 000 00000010 01000000 ........
 78 000 00000010 00100000 ........
 79 000 00000010 00010000 ........
 80 000 00000010 00001000 ........
 81 000 00000010 00000100 ........
 82 000 00000010 00000010 ........
 83 000 00000010 00000001 ........

 84 000 00000001 10000000 ........
 85 000 00000001 01000000 ........
 86 000 00000001 00100000 ........
 87 000 00000001 00010000 ........
 88 000 00000001 00001000 ........
 89 000 00000001 00000100 ........
 90 000 00000001 00000010 ........
 91 000 00000001 00000001 ........
```

APPENDIX A.2. First 40 columns of an incidence matrix in the standard form

```
                 11 11111111 22222222 2233333 333334
    123 45678901 23456789 01234567 8901234 567890

  1 011 11111111 00000000 00000000 0000000 000000
  2 110 00000000 11111111 00000000 0000000 000000
  3 101 00000000 00000000 11111111 0000000 000000

  4 100 10000000 00000000 00000000 0000000 00000.
  5 100 01000000 00000000 00000000 000000. 000000
  6 100 00100000 00000000 00000000 ....... ......
  7 100 00010000 00000000 00000000 ....... ......
  8 100 00001000 00000000 00000000 ....... ......
  9 100 00000100 00000000 00000000 ....... ......
 10 100 00000010 00000000 00000000 ....... ......
 11 100 00000001 00000000 00000000 ....... ......

 12 010 00000000 00000000 10000000 0000000 ......
 13 010 00000000 00000000 01000000 ....... 0.....
 14 010 00000000 00000000 00100000 .0..... ......
 15 010 00000000 00000000 00010000 ..0.... .0....
 16 010 00000000 00000000 00001000 ...0... ..0...
 17 010 00000000 00000000 00000100 ....0.. ...0..
 18 010 00000000 00000000 00000010 .....0. ....0.
 19 010 00000000 00000000 00000001 0...... 000000

 20 001 00000000 10000000 00000000 0000000 ......
 21 001 00000000 01000000 00000000 ....... 0.....
 22 001 00000000 00100000 00000000 0..... 000000
 23 001 00000000 00010000 00000000 .0..... .0....
 24 001 00000000 00001000 00000000 ..0.... ..0...
 25 001 00000000 00000100 00000000 ...0... ...0..
 26 001 00000000 00000010 00000000 ....0.. ....0.
 27 001 00000000 00000001 00000000 .....0. ......

 28 000 10000000 10000000 10000000 1111111 000000
 29 000 10000000 01000000 01000000 0000000 100000
 30 000 10000000 00100000 00100000 0000000 000000
 31 000 10000000 00010000 00010000 0000000 010000
 32 000 10000000 00001000 00001000 0000000 001000
 33 000 10000000 00000100 00000100 0000000 000100
 34 000 10000000 00000010 00000010 0000000 000010
 35 000 10000000 00000001 00000001 0000000 000000

 36 000 01000000 10000000 01000000 0000000 000000
 37 000 01000000 01000000 10000000 0000000 000000
 38 000 01000000 00100000 00000001 1000000 111111
 39 000 01000000 00010000 00100000 0100000 000000
 40 000 01000000 00001000 00010000 0010000 000000
 41 000 01000000 00000100 00001000 0001000 000000
 42 000 01000000 00000010 00000100 0000100 000000
 43 000 01000000 00000001 00000010 0000010 000000

 44 000 00100000 10000000 00000001 0000000 000000
 45 000 00100000 01000000 00001000 ...0... 0.0...
 46 000 00100000 00100000 10000000 0000000 000000
 47 000 00100000 00010000 00000010 .0...0. .0..0.
 48 000 00100000 00001000 00100000 .00.... ..0...
 49 000 00100000 00000100 01000000 ...0... 0..0..
 50 000 00100000 00000010 00010000 ..0.0.. .0..0.
 51 000 00100000 00000001 00000100 ....00. ...0..

 52 000 00010000 10000000 00001000 0000000 ..0...
 53 000 00010000 01000000 00000001 0..... 000000
 54 000 00010000 00100000 00000100 0...0.. 000000
 55 000 00010000 00010000 10000000 0000000 .0....
 56 000 00010000 00001000 00000010 ..0..0. ..0.0.
 57 000 00010000 00000100 00100000 .0.0... ...0..
 58 000 00010000 00000010 01000000 ....0.. 0..0.
 59 000 00010000 00000001 00010000 ..0..0. .0....

 60 000 00001000 10000000 00000010 0000000 ....0.
 61 000 00001000 01000000 00010000 ..0.... 00....
 62 000 00001000 00100000 01000000 0..... 000000
 63 000 00001000 00010000 00000100 .0..0.. .0.0..
 64 000 00001000 00001000 00000001 0.0.... 000000
 65 000 00001000 00000100 10000000 0000000 ...0..
 66 000 00001000 00000010 00100000 .0..0.. ....0.
 67 000 00001000 00000001 00001000 ...0.0. ..0...

 68 000 00000100 10000000 00010000 0000000 .0....
 69 000 00000100 01000000 00000010 .....0. 0. .0.
 70 000 00000100 00100000 00001000 0..0... 000000
 71 000 00000100 00010000 01000000 .0..... 00....
 72 000 00000100 00001000 00000100 ..0.0.. ..00..
 73 000 00000100 00000100 00000001 0..0... 000000
 74 000 00000100 00000010 10000000 0000000 ....0.
 75 000 00000100 00000001 00100000 .0....0. ......

 76 000 00000010 10000000 00100000 0000000 ......
 77 000 00000010 01000000 00000100 ....0.. 0..0..
 78 000 00000010 00100000 00000010 0....0. 000000
 79 000 00000010 00010000 00001000 .0.0... .00...
 80 000 00000010 00001000 10000000 0000000 ..0...
 81 000 00000010 00000100 00010000 ..00... .0.0..
 82 000 00000010 00000010 00000001 0...0.. 000000
 83 000 00000010 00000001 01000000 .....0. 0.....

 84 000 00000001 10000000 00000100 0000000 ...0..
 85 000 00000001 01000000 00100000 .0..... 0.....
 86 000 00000001 00100000 00010000 0.0.... 000000
 87 000 00000001 00010000 00000001 00..... 000000
 88 000 00000001 00001000 01000000 ..0.... 0.0...
 89 000 00000001 00000100 00030010 ...0.0. ...00.
 90 000 00000001 00000010 00001000 ...00.. ..0.0.
 91 000 00000001 00000001 10000000 0000000 ......
```

APPENDIX B. List of 21 latin squares which extend to complete planes

Latin square	# of extensions to 40 columns	Planes	# of copies	|Aut| conjugacy	|Aut| isotopy
1::	1 12345678 21456783 34167825 47583216 56728134 78631542 85274361 63812457	right	1	6	1
2::	1 12345678 21456783 34167825 48573261 56718432 67281354 73824516 85632147	left	1	6	1
3::	1 12345678 21456783 34167825 56781342 48623157 67238514 75814236 83572461	Hughes	1	2	3
4::	4 12345678 21456783 34167825 68572134 56218347 73684251 47823516 85731462	Hughes	4	2	6

5::	1 12345678 21456783 34167825 78531462 85672134 46823517 57218346 63784251	Hughes	1	2	2
6::	2 12345678 21456783 34172865 53681427 48567231 75218346 86723154 67834512	left, right	1,1	2	6
7::	1 12345678 21456783 34172865 57861234 63284517 48537126 75618342 86723451	Hughes	1	1	1
8::	1 12345678 21456783 34267851 73182564 56823417 85734126 48671235 67518342	Hughes	1	6	1
9::	1 12345678 21456783 34271856 56814327 78162534 83627415 45783162 67538241	Hughes	1	6	3

10::	1 12345678 21456783 34572816 53814267 86723154 68237541 47681325 75168432	Hughes	1	1	2
11::	1 12345678 21456783 34627815 47568321 58273146 76812534 63184257 85731462	Hughes	1	2	3
12::	6 12345678 21456783 34678215 47582361 65731842 86217534 53864127 78123456	left, right	3,3	6	6
13::	1 12345678 21456783 34712856 67583412 43168527 58674231 75821364 86237145	Hughes	1	6	1
14::	1 12345678 21456783 34712856 68573142 43867215 56184327 75238461 87621534	Hughes	1	2	2

15::	1 12345678 21456783 35287164 47128536 53614827 68732415 74861352 86573241	Hughes	1	1	12
16::	3 12345678 21456783 36572841 47681352 63728415 74813526 58164237 85237164	left, right, none	1,1,1	2	48
17::	8 12345678 21456783 36572841 58617324 74268135 47823516 63184257 85731462	Hughes	8	2	192
18::	1 12345678 21456783 34568217 43781526 67214835 76823451 58172364 85637142	Hughes	1	2	6
19::	8 12345678 23416785 34567812 41678523 67852341 78123456 85234167 56781234	Hughes	8	2	128

20::	280 12345678 23416785 41238567 58763214 65874321 76581432 87652143 34127856	left, right, Hughes	108,108,64	6	1536
21::	2 12345678 23416785 41238567 34127856 56782341 67853412 78564123 85671234	Desarguesian	2	6	256

APPENDIX C. 91x40 with no extension to a complete matrix.

```
                  11 11111111 22222222 2233333 333334 444444 44455555555556666666666777777777789888888899
   123 45678901 23456789 01234567 8901234 567890 123456 78901234567890123456789012345678901234567890123456789012345678901

  1 011 11111111 00000000 00000000 0000000 000000 000000 000000000000000000000000000000000000000000000000000000000000000
  2 110 00000000 11111111 00000000 0000000 000000 000000 000000000000000000000000000000000000000000000000000000000000000
  3 101 00000000 00000000 11111111 0000000 000000 000000 000000000000000000000000000000000000000000000000000000000000000

  4 100 10000000 00000000 00000000 0000000 000001 ..... .....................................................
  5 100 01000000 00000000 00000000 0000001 000000 ..... .....................................................
  6 100 00100000 00000000 00000000 1000000 000000 ..... .....................................................
  7 100 00010000 00000000 00000000 0000010 000000 ..... .....................................................
  8 100 00001000 00000000 00000000 0100000 000010 ..... .....................................................
  9 100 00000100 00000000 00000000 0001000 100000 ..... .....................................................
 10 100 00000010 00000000 00000000 0000100 000100 ..... .....................................................
 11 100 00000001 00000000 00000000 0010000 001000 ..... .....................................................

 12 010 00000000 00000000 10000000 0000000 000010 ..... .....................................................
 13 010 00000000 00000000 01000000 0000001 000100 ..... .....................................................
 14 010 00000000 00000000 00100000 1000000 000000 ..... .....................................................
 15 010 00000000 00000000 00010000 0000010 100000 ..... .....................................................
 16 010 00000000 00000000 00001000 0010000 000001 ..... .....................................................
 17 010 00000000 00000000 00000100 0100000 010000 ..... .....................................................
 18 010 00000000 00000000 00000010 0000100 001000 ..... .....................................................
 19 010 00000000 00000000 00000001 0001000 000000 ..... .....................................................

 20 001 00000000 10000000 00000000 0000000 010000 ..... .....................................................
 21 001 00000000 01000000 00000000 0000001 001000 ..... .....................................................
 22 001 00000000 00100000 00000000 0001000 000000 ..... .....................................................
 23 001 00000000 00010000 00000000 0010000 000100 ..... .....................................................
 24 001 00000000 00001000 00000000 0000010 000010 ..... .....................................................
 25 001 00000000 00000100 00000000 0000100 000001 ..... .....................................................
 26 001 00000000 00000010 00000000 0100000 100000 ..... .....................................................
 27 001 00000000 00000001 00000000 1000000 000000 ..... ..... .....................................................

 28 000 10000000 10000000 10000000 1111111 000000 000000 000000000000000000000000000000000000000000000000000000000000000
 29 000 10000000 01000000 01000000 0000000 100000 ..... .....................................................
 30 000 10000000 00100000 00100000 0000000 000000 ..... .....................................................
 31 000 10000000 00010000 00010000 0000000 010000 ..... .....................................................
 32 000 10000000 00001000 00001000 0000000 001000 ..... .. .....................................................
 33 000 10000000 00000100 00000100 0000000 000100 ..... .....................................................
 34 000 10000000 00000010 00000010 0000000 000010 ..... ..... .....................................................
 35 000 10000000 00000001 00000001 0000000 000000 ..... ..... ..... .....................................................

 36 000 01000000 10000000 01000000 0000000 000000 ..... .....................................................
 37 000 01000000 01000000 10000000 0000000 000000 ..... .....................................................
 38 000 01000000 00100000 00000001 1000000 111111 000000 000000000000000000000000000000000000000000000000000000000000000
 39 000 01000000 00010000 00100000 0100000 000000 ..... .....................................................
 40 000 01000000 00001000 00010000 0010000 000000 ..... .....................................................
 41 000 01000000 00000100 00001000 0001000 000000 ..... .....................................................
 42 000 01000000 00000010 00000100 0000100 000000 ..... .....................................................
 43 000 01000000 00000001 00000010 0000010 000000 ..... .....................................................

 44 000 00100000 10000000 00000001 0000000 000000 ..... .....................................................
 45 000 00100000 01000000 00001000 0100000 000100 ..... .....................................................
 46 000 00100000 00100000 10000000 0000000 000000 ..... .....................................................
 47 000 00100000 00010000 00000010 0000001 100000 ..... .....................................................
 48 000 00100000 00001000 00100000 0000100 010000 ..... .....................................................
 49 000 00100000 00000100 01000000 0000010 001000 ..... .....................................................
 50 000 00100000 00000010 0001u000 0001000 000001 ..... .....................................................
 51 000 00100000 00000001 00000100 0010000 000010 ..... .....................................................

 52 000 00010000 10000000 00001000 0000000 000010 ..... . .....................................................
 53 000 00010000 01000000 00000001 0000100 000000 ..... .....................................................
 54 000 00010000 00100000 00000100 0000001 000000 ..... .....................................................
 55 000 000x0000 00010000 10000000 0000000 000001 ..... .....................................................
 56 000 00010000 00001000 00000010 0001000 000100 ..... .....................................................
 57 000 00010000 00000100 00100000 0010000 100000 ..... .....................................................
 58 000 00010000 00000010 01000000 1000000 000000 ..... .....................................................
 59 000 00010000 00000001 00010000 0100000 001000 ..... .....................................................

 60 000 00001000 10000000 00000010 0000000 000001 ..... .....................................................
 61 000 00001000 01000000 0001000  1000000 000000 ..... .....................................................
 62 000 00001000 00100000 01000000 0010000 000000 ..... .....................................................
 63 000 00001000 00010000 00000100 0001000 001000 ..... .....................................................
 64 000 00001000 00001000 00000001 0000001 000000 ..... .....................................................
 65 000 00001000 00000100 10000000 0000000 010000 . .... .....................................................
 66 000 00001000 00000010 00100000 0000010 000100 ..... .....................................................
 67 000 00001000 00000001 00001000 0000100 100000 ..... .....................................................

 68 000 00000100 10000000 00010000 0000000 000100 ..... .....................................................
 69 000 00000100 01000000 0000001? ?010000 010000 ..... .....................................................
 70 000 00000100 00100000 0000100. . 00010 000000 ..... .....................................................
 71 000 00000100 00010000 01000000 0000100 000010 ..... .....................................................
 72 000 00000100 00001000 00000100 1000000 000000 ..... ..... .....................................................
 73 000 00000100 00000100 00000001 0100000 000000 ..... .....................................................
 74 000 00000100 00000010 10000000 0000000 001000 ..... .....................................................
 75 000 00000100 00000001 00100000 0000001 000001 ..... .....................................................

 76 000 00000010 10000000 00100000 0000000 001000 ..... .....................................................
 77 000 00000010 01000000 00000100 0000010 000001 ..... .....................................................
 78 000 00000010 00100000 00000010 0100000 000000 ..... .....................................................
 79 000 00000010 00010000 00001000 1000000 000000 ..... .....................................................
 80 000 00000010 00001000 10000000 0000000 100000 ..... .....................................................
 81 000 00000010 00000100 00010000 0000001 000010 ..... .....................................................
 82 000 00000010 00000010 00000001 0010000 000000 ..... .....................................................
 83 000 00000010 00000001 01000000 0001000 010000 ..... .....................................................

 84 000 00000001 10000000 00000100 0000000 100000 ..... .....................................................
 85 000 00000001 01000000 00100000 0001000 000010 ..... .....................................................
 86 000 00000001 00100000 00010000 0000100 000000 ..... .....................................................
 87 000 00000001 00010000 00000001 0000010 000000 ..... .....................................................
 88 000 00000001 00001000 01000000 0100000 000001 ..... .....................................................
 89 000 00000001 00000100 00000010 1000000 000000 ..... .....................................................
 90 000 00000001 00000010 00001000 0000001 010000 ..... .....................................................
 91 000 00000001 00000001 10000000 0000000 000100 ..... .....................................................
```

APPENDIX D.1. The Desarguesian plane.

```
                 11 11111111 22222222 22333333 33334444 44444455 55555555 66666666 66777777 77778888 88888899
    123 45678901 23456789 01234567 89012345 67890123 45678901 23456789 01234567 89012345 67890123 45678901

  1 011 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  2 110 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  3 101 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

  4 100 10000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  5 100 01000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000
  6 100 00100000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000
  7 100 00010000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000
  8 100 00001000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000
  9 100 00000100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000
 10 100 00000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000
 11 100 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111

 12 010 00000000 00000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000
 13 010 00000000 00000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000
 14 010 00000000 00000000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000
 15 010 00000000 00000000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000
 16 010 00000000 00000000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000
 17 010 00000000 00000000 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100
 18 010 00000000 00000000 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010
 19 010 00000000 00000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001

 20 001 00000000 10000000 00000000 00100000 01000000 10000000 00010000 00000100 00001000 00000001 00000010
 21 001 00000000 01000000 00000000 00010000 00100000 01000000 10000000 00000010 00000100 00001000 00000001
 22 001 00000000 00100000 00000000 10000000 00010000 00100000 01000000 00000001 00000010 00000100 00001000
 23 001 00000000 00010000 00000000 01000000 10000000 00010000 00100000 00001000 00000001 00000010 00000100
 24 001 00000000 00001000 00000000 00000010 00000100 00001000 00010000 00000001 00100000 01000000 10000000
 25 001 00000000 00000100 00000000 00000001 00000010 00000100 00001000 00010000 00100000 01000000 10000000
 26 001 00000000 00000010 00000000 00001000 00000001 00000010 00000100 10000000 00010000 00100000 01000000
 27 001 00000000 00000001 00000000 00000100 00001000 00000001 00000010 01000000 10000000 00010000 00100000

 28 000 10000000 10000000 10000000 00000000 00000100 00100000 00001000 00000010 00000001 00010000 01000000
 29 000 10000000 01000000 01000000 00000000 00000010 00010000 00000100 00000001 00001000 10000000 00100000
 30 000 10000000 00100000 00100000 00000000 00000001 10000000 00000010 00001000 00000100 01000000 00010000
 31 000 10000000 00010000 00010000 00000000 00001000 01000000 00000001 00000100 00000010 00100000 10000000
 32 000 10000000 00001000 00001000 00000000 00100000 00000010 01000000 00010000 10000000 00000001 00000100
 33 000 10000000 00000100 00000100 00000000 00010000 00000001 00100000 10000000 01000000 00001000 00000010
 34 000 10000000 00000010 00000010 00000000 10000000 00001000 00010000 01000000 00100000 00000100 00000001
 35 000 10000000 00000001 00000001 00000000 01000000 00000100 10000000 00100000 00010000 00000010 00001000

 36 000 01000000 10000000 00010000 00000001 00000000 00001000 01000000 10000000 00000100 00000010 00100000
 37 000 01000000 01000000 10000000 00001000 00000000 00000100 00100000 01000000 00000010 00000001 00010000
 38 000 01000000 00100000 01000000 00000100 00000000 00000010 00010000 00100000 00000001 00001000 10000000
 39 000 01000000 00010000 00100000 00000010 00000000 00000001 10000000 00010000 00001000 00000100 01000000
 40 000 01000000 00001000 00000001 10000000 00000000 01000000 00000100 00001000 00100000 00010000 00000010
 41 000 01000000 00000100 00001000 01000000 00000000 00100000 00000010 10000100 00010000 10000000 00000001
 42 000 01000000 00000010 00000100 00100000 00000000 00010000 00000001 00000010 10000000 01000000 00001000
 43 000 01000000 00000001 00000010 00010000 00000000 10000000 00001000 00000001 01000000 00100000 00000100

 44 000 00100000 10000000 00100000 10000000 00000010 00000000 00000001 01000000 00010000 00001000 00000100
 45 000 00100000 01000000 00010000 01000000 00000001 00000000 00001000 00100000 10000000 00000100 00000010
 46 000 00100000 00100000 10000000 00100000 00001000 00000000 00000100 00010000 01000000 00000010 00000001
 47 000 00100000 00010000 01000000 00010000 00000100 00000000 00000010 10000000 00100000 00000001 00001000
 48 000 00100000 00001000 00000010 00001000 00010000 00000000 10000000 00000100 00000001 01000000 00100000
 49 000 00100000 00000100 00000001 00000100 10000000 00000000 01000000 00000010 00001000 00100000 00010000
 50 000 00100000 00000010 00001000 00000010 01000000 00000000 00100000 00000001 00000100 00010000 10000000
 51 000 00100000 00000001 00000100 00000001 00100000 00000000 00010000 00001000 00000010 10000000 01000000

 52 000 00010000 10000000 01000000 00000010 00010000 00000100 00000000 00001000 10000000 00100000 00000001
 53 000 00010000 01000000 00100000 00000001 10000000 00000010 00000000 00000100 01000000 00010000 00001000
 54 000 00010000 00100000 00010000 00001000 01000000 00000001 00000000 00000010 00100000 10000000 00000100
 55 000 00010000 00010000 10000000 00000100 00100000 00001000 00000000 00000001 00010000 01000000 00000010
 56 000 00010000 00001000 00000100 00010000 00000001 00100000 00000000 01000000 00001000 00000010 10000000
 57 000 00010000 00000100 00000010 10000000 00001000 00010000 00000000 00100000 00000100 00000001 01000000
 58 000 00010000 00000010 00000001 01000000 00000100 10000000 00000000 00010000 00000010 00001000 00100000
 59 000 00010000 00000001 00001000 00100000 00000010 01000000 00000000 10000000 00000001 00000100 00010000

 60 000 00001000 10000000 00000001 00001000 00100000 00010000 00000010 00000000 01000000 00000100 10000000
 61 000 00001000 01000000 00001000 00000100 00010000 10000000 00000001 00000000 00100000 00000010 01000000
 62 000 00001000 00100000 00000100 00000010 10000000 01000000 00001000 00000000 00010000 00000001 00100000
 63 000 00001000 00010000 00000010 00000001 01000000 00100000 00000100 00000000 10000000 00001000 00010000
 64 000 00001000 00001000 10000000 01000000 00000010 00000001 00000010 00000000 00001000 00100000 00001000
 65 000 00001000 00000100 01000000 00100000 00000001 00001000 10000000 00000000 00000010 00010000 00000100
 66 000 00001000 00000010 00100000 00010000 00001000 00000100 01000000 00000000 00000001 10000000 00000010
 67 000 00001000 00000001 00010000 10000000 00000100 00000010 00100000 00000000 00001000 01000000 00000001

 68 000 00000100 10000000 00000010 00000100 00000001 01000000 00100000 00010000 00000000 10000000 00001000
 69 000 00000100 01000000 00000001 00000010 00001000 00100000 00010000 10000000 00000000 01000000 00000100
 70 000 00000100 00100000 00001000 00000001 00000100 00010000 10000000 01000000 00000000 00100000 00000010
 71 000 00000100 00010000 00000100 00001000 00000010 10000000 01000000 00100000 00000000 00010000 00000001
 72 000 00000100 00001000 00010000 00100000 10000000 00000100 00000010 00000001 00000000 00001000 01000000
 73 000 00000100 00000100 10000000 00010000 01000000 00000010 00000001 00001000 00000000 00000100 00100000
 74 000 00000100 00000010 01000000 10000000 00100000 00000001 00001000 00000100 00000000 00000010 00010000
 75 000 00000100 00000001 00100000 01000000 00010000 00001000 00000100 00000010 00000000 00000001 10000000

 76 000 00000010 10000000 00000100 01000000 00001000 00000010 10000000 00000001 00100000 00000000 00010000
 77 000 00000010 01000000 00000010 00100000 00000100 00000001 01000000 00001000 00010000 00000000 10000000
 78 000 00000010 00100000 00000001 00010000 00000010 00001000 00100000 00000100 10000000 00000000 01000000
 79 000 00000010 00010000 00001000 10000000 00000001 00000100 00010000 00000010 01000000 00000000 00100000
 80 000 00000010 00001000 00100000 00000100 01000000 00010000 00001000 10000000 00000010 00000000 00000001
 81 000 00000010 00000100 00010000 00000010 00100000 10000000 00000100 01000000 00000001 00000000 00001000
 82 000 00000010 00000010 10000000 00000001 00010000 01000000 00000010 00100000 00001000 00000000 00000100
 83 000 00000010 00000001 01000000 00001000 10000000 00100000 00000001 00010000 00000100 00000000 00000010

 84 000 00000001 10000000 00001000 00010000 10000000 00000001 00000100 00100000 00000010 01000000 00000000
 85 000 00000001 01000000 00000100 10000000 01000000 00001000 00000010 00010000 00000001 00100000 00000000
 86 000 00000001 00100000 00000010 01000000 00100000 00000100 00000001 10000000 00001000 00010000 00000000
 87 000 00000001 00010000 00000001 00100000 00010000 00000010 00001000 01000000 00000100 10000000 00000000
 88 000 00000001 00001000 01000000 00000001 00001000 10000000 00100000 00000010 00010000 00000100 00000000
 89 000 00000001 00000100 00100000 00001000 00000100 00100000 00010000 00000001 10000000 00000010 00000000
 90 000 00000001 00000010 00010000 00000100 00000010 00100000 10000000 00001000 01000000 00000001 00000000
 91 000 00000001 00000001 10000000 00000010 00000001 00010000 01000000 00000100 00100000 00001000 00000000
```

APPENDIX D.2. The Hughes plane.

```
                 11 11111111 22222222 22333333 33334444 44444455 55555555 66666666 66777777 77778888 88888899
    123 45678901 23456789 01234567 89012345 67890123 45678901 23456789 01234567 89012345 67890123 45678901

  1 011 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  2 110 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  3 101 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

  4 100 10000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  5 100 01000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000
  6 100 00100000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000
  7 100 00010000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000
  8 100 00001000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000
  9 100 00000100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000
 10 100 00000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000
 11 100 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111

 12 010 00000000 00000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000
 13 010 00000000 00000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000
 14 010 00000000 00000000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000
 15 010 00000000 00000000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000
 16 010 00000000 00000000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000
 17 010 00000000 00000000 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100
 18 010 00000000 00000000 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010
 19 010 00000000 00000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001

 20 001 00000000 10000000 00000000 00010000 01000000 00100000 10000000 00000010 00000001 00001000 00000100
 21 001 00000000 01000000 00000000 00000100 00100000 00000001 01000000 00010000 00001000 10000000 00000010
 22 001 00000000 00100000 00000000 00000010 00001000 01000000 00100000 00000100 10000000 00000001 00010000
 23 001 00000000 00010000 00000000 10000000 00000100 00000010 00010000 00000001 01000000 00100000 00001000
 24 001 00000000 00001000 00000000 00000001 00000010 00010000 00001000 00100000 00000100 01000000 10000000
 25 001 00000000 00000100 00000000 01000000 00000001 10000000 00000100 00001000 00000010 00010000 00100000
 26 001 00000000 00000010 00000000 00100000 10000000 00001000 00000010 01000000 00010000 00000100 00000001
 27 001 00000000 00000001 00000000 00001000 00010000 00000100 00000001 10000000 00100000 00000010 01000000

 28 000 10000000 10000000 10000000 00000000 00100000 01000000 00010000 00001000 00000100 00000010 00000001
 29 000 10000000 01000000 01000000 00000000 10000000 00010000 00000100 00000010 00100000 00000001 00001000
 30 000 10000000 00100000 00100000 00000000 00010000 10000000 00000010 00000001 00001000 01000000 00000100
 31 000 10000000 00010000 00010000 00000000 00001000 00000001 10000000 00100000 00000010 00000100 01000000
 32 000 10000000 00001000 00001000 00000000 00000100 00000010 00000001 00000100 00010000 10000000 00100000
 33 000 10000000 00000100 00000100 00000000 00000010 00001000 01000000 10000000 00000001 00100000 00010000
 34 000 10000000 00000010 00000010 00000000 00000001 00000100 00100000 00010000 01000000 00001000 10000000
 35 000 10000000 00000001 00000001 00000000 00000100 00100000 00001000 01000000 10000000 00010000 00000010

 36 000 01000000 10000000 01000000 00100000 00000000 00000100 00001000 00000001 00000010 10000000 00010000
 37 000 01000000 01000000 10000000 00000010 00000000 00001000 00000001 00100000 01000000 00010000 00000100
 38 000 01000000 00100000 00000001 00010000 00000000 00000010 01000000 00001000 00100000 00000100 10000000
 39 000 01000000 00010000 00100000 00000001 00000000 01000000 00000100 10000000 00010000 00001000 00000010
 40 000 01000000 00001000 00010000 00000100 00000000 10000000 00100000 01000000 00000001 00000010 00001000
 41 000 01000000 00000100 00001000 10000000 00000000 00100000 00000010 00010000 00000100 00000001 01000000
 42 000 01000000 00000010 00000100 00001000 00000000 00000001 00010000 00000010 10000000 01000000 00100000
 43 000 01000000 00000001 00000010 01000000 00000000 00000100 10000000 00000100 00001000 00100000 00000001

 44 000 00100000 10000000 00100000 01000000 00000010 00000000 00000001 00010000 10000000 00000100 00001000
 45 000 00100000 01000000 00001000 00010000 00000100 00000000 00100000 10000000 00000010 01000000 00000001
 46 000 00100000 00100000 10000000 00000100 00000001 00000000 00001000 00010000 00000010 00100000 01000000
 47 000 00100000 00010000 01000000 00001000 00100000 00000000 00000010 00000100 00000001 00010000 10000000
 48 000 00100000 00001000 00000100 00100000 00010000 00000000 10000000 00001000 01000000 00000001 00000010
 49 000 00100000 00000100 00000010 00000001 00001000 00000000 00100000 01000000 00100000 10000000 00000100
 50 000 00100000 00000010 00000001 10000000 01000000 00000000 00000100 00100000 00001000 00000010 00010000
 51 000 00100000 00000001 00010000 00000010 10000000 00000000 01000000 00000001 00000100 00001000 00100000

 52 000 00010000 10000000 00010000 10000000 00000001 00001000 00000000 00000100 00100000 01000000 00000010
 53 000 00010000 01000000 00000100 01000000 00001000 00100000 00000000 00000001 00010000 00000010 10000000
 54 000 00010000 00100000 00000010 00100000 01000000 00000001 00000000 10000000 00000100 00010000 00001000
 55 000 00010000 00010000 10000000 00010000 00000010 00000100 00000000 01000000 00001000 00000001 00100000
 56 000 00010000 00001000 00000001 00001000 10000000 01000000 00000000 00010000 00000010 00100000 00000100
 57 000 00010000 00000100 01000000 00000100 00010000 00000010 00000000 00100000 10000000 00001000 00000001
 58 000 00010000 00000010 00100000 00000010 00000100 00010000 00000000 00001000 00000001 10000000 01000000
 59 000 00010000 00000001 00001000 00000001 00100000 10000000 00000000 00000010 01000000 00000100 00010000

 60 000 00001000 10000000 00000100 00000001 10000000 00000010 00100000 00000000 00001000 00010000 01000000
 61 000 00001000 01000000 00000010 10000000 00010000 01000000 00001000 00000000 00000001 00000100 00100000
 62 000 00001000 00100000 00010000 00001000 00000010 00100000 01000000 00000100 00000000 01000000 10000000
 63 000 00001000 00010000 00001000 00100000 00000001 00010000 01000000 00000000 10000000 00000010 00000100
 64 000 00001000 00001000 10000000 01000000 00000100 00000001 00000010 00000000 00100000 00001000 00010000
 65 000 00001000 00000100 00000001 00000010 00100000 00000100 10000000 00000000 00010000 01000000 00001000
 66 000 00001000 00000010 01000000 00010000 00001000 10000000 00000001 00000000 00000100 00100000 00000010
 67 000 00001000 00000001 00100000 00000100 01000000 00001000 00010000 00000000 00000010 00000001 10000000

 68 000 00000100 10000000 00001000 00000010 00010000 00000001 00000100 01000000 00000000 00100000 10000000
 69 000 00000100 01000000 00000001 00100000 00000010 10000000 00010000 00000100 00000000 00001000 01000000
 70 000 00000100 00100000 01000000 00000001 00000100 00001000 10000000 00010000 00000000 00000010 00100000
 71 000 00000100 00010000 00000010 00000100 10000000 00100000 00000001 00001000 00000000 01000000 00010000
 72 000 00000100 00001000 00100000 10000000 00001000 00000100 01000000 00000010 00000000 00010000 00000001
 73 000 00000100 00000100 10000000 00001000 01000000 00010000 00100000 00000001 00000000 00000100 00000010
 74 000 00000100 00000010 00010000 01000000 00100000 00000010 00001000 10000000 00000000 00000001 00000100
 75 000 00000100 00000001 00000100 00010000 00000001 01000000 00000010 00100000 00000000 10000000 00001000

 76 000 00000010 10000000 00000001 00000100 00001000 00010000 00000010 10000000 01000000 00000000 00100000
 77 000 00000010 01000000 00100000 00001000 00000001 00000010 10000000 01000000 00000100 00000000 00010000
 78 000 00000010 00100000 00001000 01000000 10000000 00000100 00010000 00100000 00000001 00000000 00000010
 79 000 00000010 00010000 00000100 00000010 01000000 10000000 00001000 00010000 00100000 00000000 00000001
 80 000 00000010 00001000 00000010 00010000 00100000 00001000 00000100 00000001 10000000 00000000 01000000
 81 000 00000010 00000100 00010000 00100000 00000100 01000000 00000001 00000010 00001000 00000000 10000000
 82 000 00000010 00000010 10000000 00000001 00010000 00100000 01000000 00000100 00000010 00000000 00001000
 83 000 00000010 00000001 01000000 10000000 00000010 00000001 00100000 00001000 00010000 00000000 00000100

 84 000 00000001 10000000 00000010 00001000 00000100 10000000 01000000 00100000 00010000 00000001 00000000
 85 000 00000001 01000000 00010000 00000001 01000000 00000100 00000010 00001000 10000000 00100000 00000000
 86 000 00000001 00100000 00001000 10000000 00100000 00010000 00000001 01000000 00000100 00001000 00000000
 87 000 00000001 00010000 00000001 01000000 00010000 00001000 00100000 00000010 00000100 10000000 00000000
 88 000 00000001 00001000 01000000 00000010 00000001 00100000 00010000 10000000 00001000 00000100 00000000
 89 000 00000001 00000100 00100000 00010000 10000000 00000001 00001000 00000100 01000000 00000010 00000000
 90 000 00000001 00000010 00001000 00000100 00000010 01000000 10000000 00000001 00100000 00010000 00000000
 91 000 00000001 00000001 10000000 00100000 00001000 00000010 00000100 00010000 00000001 01000000 00000000
```

APPENDIX D.3. The left near-field plane.

```
                 11 11111111 22222222 22333333 33334444 44444455 55555555 66666666 66777777 77778888 88888899
    123 45678901 23456789 01234567 89012345 67890123 45678901 23456789 01234567 89012345 67890123 45678901

  1 011 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  2 110 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  3 101 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

  4 100 10000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  5 100 01000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000
  6 100 00100000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000
  7 100 00010000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000
  8 100 00001000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000
  9 100 00000100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000
 10 100 00000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000
 11 100 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111

 12 010 00000000 00000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000
 13 010 00000000 00000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000
 14 010 00000000 00000000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000
 15 010 00000000 00000000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000
 16 010 00000000 00000000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000
 17 010 00000000 00000000 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100
 18 010 00000000 00000000 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010
 19 010 00000000 00000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001

 20 001 00000000 10000000 00000000 00100000 00000100 10000000 00001000 00000010 00000001 00010000 01000000
 21 001 00 00000 01000000 00000000 00010000 00100000 00000010 00000001 01000000 10000000 00000100 00001000
 22 001 00000000 00100000 00000000 10000000 00000001 00100000 00000010 00010000 00001000 01000000 00000100
 23 001 00000000 00010000 00000000 01000000 00001000 00010000 10000000 00000100 00100000 00000010 00000001
 24 001 00000000 00001000 00000000 00000001 00000010 01000000 00010000 10000000 00000100 00001000 00100000
 25 001 00000000 00000100 00000000 00000010 00010000 00000100 00100000 00001000 01000000 00000001 10000000
 26 0.. 00000000 00000010 00000000 00000100 10000000 00001000 01000000 00000001 00000010 00100000 00010000
 27 001 00000000 00000001 00000000 00001000 01000000 00000001 00000100 00100000 00010000 10000000 00000010

 28 000 10000000 10000000 10000000 00000000 00001000 00100000 00010000 00000001 01000000 00000100 00000010
 29 000 10000000 01000000 01000000 00000000 10000000 00010000 00000100 00000010 00001000 00000001 00100000
 30 000 10000000 00100000 00100000 00000000 01000000 10000000 00000001 00001000 00000100 00000010 00010000
 31 000 10000000 00010000 00010000 00000000 00000010 00001000 00100000 01000000 00000001 10000000 00000100
 32 000 10000000 00001000 00001000 00000000 00010000 00000001 00000010 00000100 10000000 00100000 01000000
 33 000 10000000 00000100 00000100 00000000 00000001 01000000 10000000 00100000 00000010 00010000 00001000
 34 000 10000000 00000010 00000010 00000000 00100000 00000100 00001000 10000000 00010000 01000000 00000001
 35 000 10000000 00000001 00000001 00000000 00000100 00000010 01000000 00010000 00100000 00001000 10000000

 36 000 01000000 10000000 01000000 00000001 00000000 00000010 10000000 00001000 00010000 00100000 00000100
 37 000 01000000 01000000 10000000 01000000 00000000 00000100 00000010 00100000 00000001 00001000 00010000
 38 000 01000000 00100000 00000001 00001000 00000000 00010000 00100000 10000000 00000010 00000100 01000000
 39 000 01000000 00010000 00100000 00000100 00000000 01000000 00001000 00010000 10000000 00000001 00000010
 40 000 01000000 00001000 00010000 00000010 00000000 10000000 00000100 00000001 00100000 0100000  0001000
 41 000 01000000 00000100 00001000 00010000 00000000 00100000 01000000 00000010 00000100 1000000  0000001
 42 000 01000000 00000010 00000100 00100000 00000000 00000001 00010000 01000000 00001000 000000   1000000
 43 000 01000000 00000001 00000010 10000000 00000000 00001000 00000001 00001000 01000000 0001000  0100000

 44 000 00100000 10000000 00100000 10000000 00010000 00000000 00000100 01000000 00000010 00001000 00000001
 45 000 00100000 01000000 00000010 00000100 00000001 00000000 00010000 00001000 00100000 10000000 01000000
 46 000 00100000 00100000 10000000 00100000 00000010 00000000 01000000 00000100 00010000 00000001 00001000
 47 000 00100000 00010000 01000000 00001000 00100000 00000000 00000010 00000001 00000100 00010000 10000000
 48 000 00100000 00001000 00000001 00010000 10000000 00000000 00001000 00100000 01000000 00000010 00000100
 49 000 00100000 00000100 00010000 01000000 00000100 00000000 00000001 10000000 00001000 00100000 00000010
 50 000 0010000. 00000010 00001000 00000010 01000000 00000000 10000000 00010000 00000001 00000100 00100000
 51 000 00100000 00000001 00000100 00000001 00001000 00000000 00100000 00000010 10000000 01000000 00010000

 52 000 00010000 10000000 00000001 00000010 00100000 01000000 00000000 00000100 00001000 10000000 00100000
 53 000 00010000 01000000 00000100 00001000 00000010 10000000 00000000 00001000 01000000 00100000 00000001
 54 000 00010000 00100000 00001000 00000001 10000000 00000100 00000000 01000000 00100000 00010000 00000010
 55 000 00010000 00010000 10000000 00010000 00000100 00000001 00000000 00001000 00000010 01000000 00100000
 56 000 0001 000 00001000 00100000 01000000 00000001 00001000 00000000 00000010 00010000 00000100 10000000
 57 000 00010000 00000100 00000010 00100000 01000000 00010000 00000000 00000001 10000000 00001000 00000100
 58 000 00010000 00000010 000.0000 10000000 00001000 00000010 00000000 00100000 00000100 00000001 01000000
 59 000 00010000 00000001 0100000 00000100 00010000 00100000 00000000 10000000 00000001 00000010 00001000

 60 000 00001000 10000000 00010000 00001000 00000001 00000100 01000000 00000000 10000000 00000010 00100000
 61 000 0000'000 01000000 00000001 00100000 00001000 00000010 10000000 00000000 00001000 01000000 00000010
 62 000 00001000 00100000 00000010 00010000 00001000 01000000 00000100 00000000 00000001 00100000 10000000
 63 000 00001000 00010000 00000100 00000010 10000000 00100000 00000001 00000000 00010000 00001000 01000000
 64 000 00001000 00001000 10000000 00000100 01000000 00000010 00100000 00000000 00001000 00010000 00000001
 65 000 00001000 00000100 01000000 10000000 00000010 00000001 00001000 00000000 00100000 00000100 00010000
 66 000 00001000 00000010 00100000 00000001 00000100 00010000 00000010 00000000 01000000 10000000 00001000
 67 000 00001000 00000001 00001000 01000000 00100000 10000000 00010000 00000000 00000010 00000001 00000100

 68 000 00000100 10000000 00001000 00000100 00000010 00010000 00000001 00100000 00000000 01000000 10000000
 69 000 00000100 01000000 00100000 00000010 00001000 00000001 01000000 10000000 00000000 00010000 00000100
 70 000 00000100 00100000 00000100 01000000 00010000 00000010 00001000 00000001 00000000 10000000 00100000
 71 000 00000100 00010000 00000001 10000000 01000000 00000100 00010000 00000010 00000000 00100000 00001000
 72 000 00000100 00001000 00000010 00001000 00000100 00100000 10000000 01000000 00000000 00000001 00010000
 73 000 00000100 00000100 10000000 00000001 00100000 00001000 00000100 00010000 00000000 00000010 01000000
 74 000 00000100 00000010 01000000 00010000 00000001 10000000 00100000 00001000 00000000 00001000 00000010
 75 000 00000100 00000001 00010000 00100000 10000000 01000000 00000010 00001000 00000000 00000100 00000001

 76 000 00000010 10000000 00000010 01000000 10000000 00000001 00100000 00010000 00000100 00000000 00001000
 77 000 00000010 01000000 00010000 00000001 01000000 00100000 00001000 00000100 00000010 00000000 10000000
 78 000 00000010 00100000 00100000 00000010 00000100 01000000 00010000 00100000 10000000 00000000 00000001
 79 000 00000010 00010000 00001000 00100000 00000001 00000010 00000100 10000000 01000000 00000000 00010000
 80 000 00000010 00001000 00000100 10000000 00100000 00010000 01000000 00001000 00000001 00000000 00000010
 81 000 00000010 00000100 00000001 00000100 00001000 10000000 00000010 01000000 00010000 00000000 00100000
 82 000 00000010 00000010 10000000 00001000 00010000 01000000 00000001 00000010 00100000 00000000 00000100
 83 000 00000010 00000001 00100000 00010000 00000010 00000100 10000000 00000001 00001000 00000000 01000000

 84 000 00000001 10000000 00000100 00010000 01000000 00001000 00000010 10000000 00100000 00000001 00000000
 85 000 00000001 01000000 00001000 10000000 00000100 01000000 00100000 00000001 00010000 00000010 00000000
 86 000 00000001 00100000 00010000 00000100 00100000 00000001 10000000 00000010 01000000 00001000 00000000
 87 000 00000001 00010000 0000001C 00000001 00010000 10000000 01000000 00100000 00001000 00000100 00000000
 88 000 00000001 00001000 01000000 00100000 00001000 00000100 00000001 00010000 00000010 10000000 00000000
 89 000 00000001 00000100 00100000 00001000 10000000 00000010 00010000 00000100 00000001 01000000 00000000
 90 000 00000001 00000010 00000001 01000000 00000010 00100000 00000100 00001000 10000000 0C010000 00000000
 91 000 00000001 00000001 10000000 00000010 00000001 00010000 00001000 01000000 00000100 00100000 00000000
```

APPENDIX D.4. The right near-field plane.

```
                 11 11111111 22222222 22333333 33334444 44444455 55555555 66666666 66777777 77778888 88888899
    123 45678901 23456789 01234567 89012345 67890123 45678901 23456789 01234567 89012345 67890123 45678901

  1 011 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  2 110 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  3 101 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

  4 100 10000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  5 100 01000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000 00000000
  6 100 00100000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000 00000000
  7 100 00010000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000 00000000
  8 100 00001000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000 00000000
  9 100 00000100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000 00000000
 10 100 00000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000
 11 100 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 11111111

 12 010 00000000 00000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000 10000000
 13 010 00000000 00000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000 01000000
 14 010 00000000 00000000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000 00100000
 15 010 00000000 00000000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000 00010000
 16 010 00000000 00000000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000 00001000
 17 010 00000000 00000000 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100 00000100
 18 010 00000000 00000000 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010 00000010
 19 010 00000000 00000000 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001 00000001

 20 001 00000000 10000000 00000000 10000000 00000001 00000100 00010000 00000010 00001000 01000000 00100000
 21 001 00000000 01000000 00000000 00100000 00000010 00000001 00000100 01000000 00010000 00001000 10000000
 22 001 00000000 00100000 00000000 00000001 10000000 00001000 00000010 00000100 00100000 00010000 01000000
 23 001 00000000 00010000 00000000 01000000 00010000 10000000 00100000 00000001 00000010 00000100 00001000
 24 001 00000000 00001000 00000000 00010000 00001000 01000000 10000000 00100000 00000100 00000010 00000001
 25 001 00000000 00000100 00000000 00000010 01000000 00010000 00001000 10000000 00000001 00100000 00000100
 26 001 00000000 00000010 00000000 00001000 00000100 00100000 01000000 00010000 10000000 00000001 00000010
 27 001 00000000 00000001 00000000 00000100 00100000 00000010 00000001 00001000 01000000 10000000 00010000

 28 000 10000000 10000000 10000000 00000000 00000010 00010000 01000000 00001000 00100000 00000100 00000001
 29 000 10000000 01000000 01000000 00000000 10000000 00000100 00001000 00100000 00000010 00000001 00010000
 30 000 10000000 00100000 00100000 00000000 00000100 00000001 00010000 10000000 01000000 00000010 00001000
 31 000 10000000 00010000 00010000 00000000 00001000 00100000 00000100 00000010 00000001 10000000 01000000
 32 000 10000000 00001000 00001000 00000000 00000001 00000010 00100000 01000000 10000000 00010000 00000100
 33 000 10000000 00000100 00000100 00000000 00100000 00001000 10000000 00000001 00010000 01000000 00000010
 34 000 10000000 00000010 00000010 00000000 00010000 01000000 00000001 00000100 00001000 00100000 10000000
 35 000 10000000 00000001 00000001 00000000 01000000 10000000 00000010 00010000 00000100 00001000 00100000

 36 000 01000000 10000000 01000000 00000100 00000000 00001000 00100000 00010000 00000001 00000010 10000000
 37 000 01000000 01000000 10000000 00001000 00000000 00010000 00100000 00000001 00000100 00100000 01000000
 38 000 01000000 00100000 00000001 00000010 00000000 00100000 10000000 01000000 00001000 00000100 00010000
 39 000 01000000 00010000 00100000 10000000 00000000 01000000 00000010 00001000 00010000 00000001 00000100
 40 000 01000000 00001000 00010000 01000000 00000000 00000100 00000001 10000000 00100000 00001000 00000010
 41 000 01000000 00000100 00001000 00100000 00000000 00000001 01000000 00000100 00000010 10000000 00100000
 42 000 01000000 00000010 00000100 00100000 00000000 10000000 00001000 00000010 01000000 00010000 00000001
 43 000 01000000 00000001 00000010 00000001 00000000 00010000 00000100 00100000 10000000 01000000 00001000

 44 000 00100000 10000000 00100000 00000001 00010000 00000000 00001000 01000000 00000100 10000000 00000010
 45 000 00100000 01000000 00000010 01000000 00000100 00000000 10000000 00001000 00000001 00010000 00100000
 46 000 00100000 00100000 10000000 00100000 00001000 00000000 00000001 00010000 00000010 01000000 00000100
 47 000 00100000 00010000 01000000 00000010 00100000 00000000 00010000 00000100 10000000 00001000 00000001
 48 000 00100000 00001000 00000001 00000100 10000000 00000000 01000000 00000010 00010000 00100000 00001000
 49 000 00100000 00000100 00010000 00001000 00000001 00000000 00000010 00100000 01000000 00000100 10000000
 50 000 00100000 00000010 00001000 10000000 01000000 00000000 00000100 00000001 00100000 00000010 00010000
 51 000 00100000 00000001 00000100 00010000 00000010 00000000 00100000 10000000 00001000 00000001 01000000

 52 000 00010000 10000000 00000010 00010000 10000000 00100000 00000000 00000001 01000000 00001000 00000100
 53 000 00010000 01000000 00000100 00000010 00000001 01000000 00000000 00010000 00100000 10000000 00001000
 54 000 00010000 00100000 00001000 01000000 00100000 00010000 00000000 00000010 00000100 00000001 10000000
 55 000 00010000 00010000 10000000 00000100 01000000 00000001 00000000 00100000 00001000 00010000 00000010
 56 000 00010000 00001000 00100000 00001000 00000010 10000000 00000000 00000100 00000001 01000000 00010000
 57 000 00010000 00000100 00000001 00100000 00010000 00000100 00000000 00001000 10000000 00000010 01000000
 58 000 00010000 00000010 01000000 00000001 00001000 00000010 00000000 10000000 00010000 00000100 00100000
 59 000 00010000 00000001 00010000 10000000 00000100 00001000 00000000 01000000 00000010 00100000 00000001

 60 000 00001000 10000000 00000100 01000000 00001000 00000001 00000010 00000000 10000000 00100000 00010000
 61 000 00001000 01000000 00010000 00000001 00100000 10000000 01000000 00000000 00001000 00000010 00000100
 62 000 00001000 00100000 00000010 00001000 01000000 00000100 00100000 00000000 00010000 10000000 00000001
 63 000 00001000 00010000 00000001 00010000 00000100 00000010 00001000 00000000 00100000 01000000 10000000
 64 000 00001000 00001000 10000000 00000010 00010000 00001000 00000100 00000000 01000000 00000001 00100000
 65 000 00001000 00000100 01000000 10000000 00000010 00100000 00000001 00000000 00000100 00010000 00001000
 66 000 00001000 00000010 00100000 00000100 00000001 00010000 10000000 00000000 00000010 00001000 01000000
 67 000 00001000 00000001 00001000 00100000 10000000 01000000 00100000 00000000 00000001 00000100 00000010

 68 000 00000100 10000000 00010000 00100000 01000000 00000010 10000000 00000100 00000000 00000001 00001000
 69 000 00000100 01000000 00001000 00000100 00010000 00100000 00000010 10000000 00000000 01000000 00000001
 70 000 00000100 00100000 01000000 00010000 00000001 10000000 00000100 00001000 00000000 00100000 00000010
 71 000 00000100 00010000 00000100 00001000 10000000 00100000 00000001 01000000 00000000 00010000 00100000
 72 000 00000100 00001000 00000010 10000000 00100000 00000001 00001000 00010000 00000000 00000100 01000000
 73 000 00000100 00000100 10000000 00000001 00000100 01000000 00100000 00000010 00000000 00001000 00010000
 74 000 00000100 00000010 00000001 01000000 00000010 00001000 00010000 00100000 00000000 10000000 00000100
 75 000 00000100 00000001 00100000 00000010 00001000 00000100 01000000 00000001 00000000 00010000 10000000

 76 000 00000010 10000000 00001000 00000010 00000100 10000000 00000001 00100000 00010000 00000000 01000000
 77 000 00000010 01000000 00000001 10000000 00001000 00100000 00000100 00010000 01000000 00000000 00100000
 78 000 00000010 00100000 00010000 00000100 00000010 01000000 00001000 00000001 10000000 00000000 00100000
 79 000 00000010 00010000 00000010 01000000 00000001 00100000 00001000 00000100 01000000 00000000 00010000
 80 000 00000010 00001000 00000100 00000001 01000000 00100000 00010000 00001000 00000010 00000000 10000000
 81 000 00000010 00000100 00100000 01000000 10000000 00000010 00000100 00010000 00001000 00000000 00000001
 82 000 00000010 00000010 10000000 00100000 00100000 00000100 00000010 01000000 00000001 00000000 00001000
 83 000 00000010 00000001 01000000 00001000 00010000 00000001 10000000 00000010 00100000 00000000 00000100

 84 000 00000001 10000000 00000001 00001000 00100000 01000000 00000100 10000000 00000010 00010000 00000000
 85 000 00000001 01000000 00100000 00010000 01000000 00001000 00000001 00000010 10000000 00000100 00000000
 86 000 00000001 00100000 00000100 10000000 00010000 00000010 01000000 00100000 00000001 00001000 00000000
 87 000 00000001 00010000 00001000 00000001 00000010 00000100 10000000 00010000 01000000 00100000 00000000
 88 000 00000001 00001000 01000000 00100000 00000100 00010000 00000010 00000001 00001000 10000000 00000000
 89 000 00000001 00000100 00000010 00000100 00001000 10000000 00010000 01000000 00100000 00000001 00000000
 90 000 00000001 00000010 00010000 00000010 10000000 00001000 00100000 00000001 00000100 01000000 00000000
 91 000 00000001 00000001 10000000 01000000 00000001 00100000 00001000 00000100 00010000 00000010 00000000
```